Universidad de Alcalá
Escuela Politécnica Superior

Máster Universitario en Ingeniería de Telecomunicación

Trabajo Fin de Máster

Sistema de Estimación de Pose de Objetos: Aplicación a Sistemas de Video Vigilancia y de Interacción Hombre Máquina

Autor: Mario García Montero

Tutor/es: Roberto López Sastre

2016

Universidad de Alcalá

Escuela Politécnica Superior

Máster en Ingeniería de Telecomunicación

Trabajo Fin de Máster

Sistema de estimación de pose de objetos: aplicación a sistemas de video vigilancia y de interacción hombre máquina

Autor: Mario García Montero

Director: Roberto López Sastre

TRIBUNAL:

Presidente: D. Hilario Gómez Moreno

Vocal 1º: Dª. Cristina Losada Gutiérrez

Vocal 2º: D. Roberto López Sastre

Sistema de estimación de pose de objetos: aplicación a sistemas de video vigilancia y de interacción hombre máquina

Mario García Montero

4 de septiembre de 2016

Te lo dedico a ti abuela, que siempre estás conmigo, por haberme empujado hacia delante en todo momento y haber hecho de esto una realidad.Te quiero.

Agradecimientos

Ante todo quiero agradecer a Roberto el haber confiado en mi para la realización de este proyecto, así como a todos los compañeros del grupo de investigación GRAM por el apoyo que me han dado en todo momento. Gracias a mis padres por haber hecho posible que haya conseguido llegar hasta aquí.

Índice general

Lista de figuras

Lista de tablas

Resumen

El término pose del objeto se puede definir como una orientación que caracteriza a un determinado objeto, y que parte de un punto de referencia ubicado en un punto estratégico dentro de dicho objeto, ya sea el centro del objeto o una zona característica. La estimación de pose que se realiza en este trabajo, es aplicada tanto a cabezas como a vehículos, para su posible integración en sistemas de interacción hombre –máquina y de videovigilancia, respectivamente. En concreto, a través de la implementación de este proyecto, se posibilita la inclusión del sistema diseñado en: 1)un sistema de monitorización constante de la pose de un usuario; y 2)en un sistema de videovigilancia de carreteras y autovías, proporcionando una división clara de los carriles de circulación, lo cual facilita tareas de conteo y análisis del flujo de vehículos que atraviesan un tramo de carretera en un instante concreto. El proyecto proporciona una serie de resultados cualitativos y cuantitativos, procedentes del análisis de imágenes de personas y de tráfico, que muestran la eficiencia del sistema diseñado en términos de velocidad y precisión y que certifican su posible incorporación a sistemas comerciales innovadores.

Palabras clave: Detección de objetos, estimación de pose, random forest, sistemas de transporte inteligente, sistemas de interacción hombre máquina.

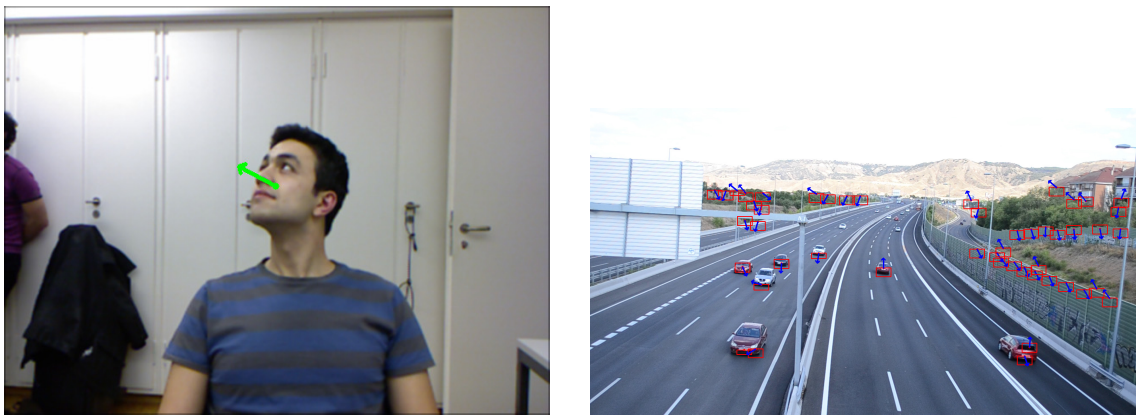
Abstract

The object pose term can be defined as an orientation that characterizes a particular object and which starts from a reference point located at a strategic point within the object, either the centre of the object or a distinctive area. In this project, pose estimation is applied to heads and vehicles, with the aim of integrating the solutions into human-machine interaction systems and video surveillance systems, respectively. In particular, through the implementation of this project, we have designed two solutions: 1)a system for constant pose monitoring of head; and 2)a road and highway monitoring system, which provides a clear division of the traffic lanes to realize vehicle counting tasks. The project provides a set of qualitative and quantitative results for the two applications described, showing the efficiency of the designed system in terms of both speed and accuracy.

Keywords: Object detection, pose estimation, random forest, smart transport systems, human-machine interaction systems.

Resumen Extendido

El desarrollo de este Trabajo Fin de Máster ha consistido en la construcción y adaptación de un sistema de estimación de pose para rostros y vehículos, haciendo uso de Hough Forests (HF). Aunque el estimador funciona a pleno rendimiento tanto en imágenes de profundidad como RGB, el trabajo desempeñado se ha centrado en optimizar el rendimiento de la estimación en imágenes RGB, lo cual conlleva un mayor reto en términos de precisión y velocidad. En la Figura 1 se muestran ejemplos de estimación tanto en cabezas humanas como en imágenes de tráfico, indicándose centro y orientación de la estimación para los dos tipos de objetos.



(a) **(b)**

Figura 1: Estimación de pose realizada en: a)Cabeza. b)Imagen de tráfico.

Las imágenes empleadas para la estimación de cabeza han sido obtenidas a través del sensor de imagen Kinect, así como de la base de datos de imágenes Biwi [9]. En cuanto a las imágenes de tráfico, se han empleado imágenes proporcionadas por cámaras de la DGT, así como imágenes procedentes de una base de datos de fotogramas de tráfico, generada específicamente para este trabajo. Así pues, el trabajo ha consistido en el ajuste de un demostrador de estimación de pose en cabezas, mediante un sistema de tracking, para posibilitar la implementación del demostrador en campos de aplicación reales para conseguir una mejor interacción hombre-máquina. Además, se ha desarrollado una solución para la estimación de orientación de vehículos para formar un sistema mas complejo de estimación de carriles que posibilite automatizar tareas de videovigilancia.

Los sistemas permiten al usuario hacer uso de las soluciones propuestas a modo demostrador en tiempo real haciendo uso de cualquier sensor de imagen del mercado. A su vez, el usuario puede testear el rendimiento del sistema a través de su ejecución , incluyendo en la llamada a la misma, secuencias de imágenes RGB almacenadas de forma local.

El sistema de estimación permite salvar cada fotograma que haya procesado con la estimación representada a modo de flecha en cada imagen. Además, permite el guardado de la secuencia de imágenes estimadas en un fichero de video, a elección del usuario.

En cuanto a la estimación de pose de cabeza, el sistema permite la representación de la pose estimada, la pose real y la pose estimada por el sistema de tracking, para comprobar los efectos de mejora introducidos en el sistema por dicho bloque de tracking. A su vez, por consola se imprimen los datos numéricos de centro y orientación (dirección, elevación y alabeo) en los 3 casos para poder analizar la diferencia entre todas las poses representadas de forma precisa. Un ejemplo visual de la triple representación de pose en cabezas puede verse en la Figura 2.

En relación a la estimación de vehículos, tan solo se realiza la representación de la pose estimada por el sistema, así como de la caja en cuyo interior se encuentra contenido el vehículo, tal y como se ha podido ver en la figura 1(b).

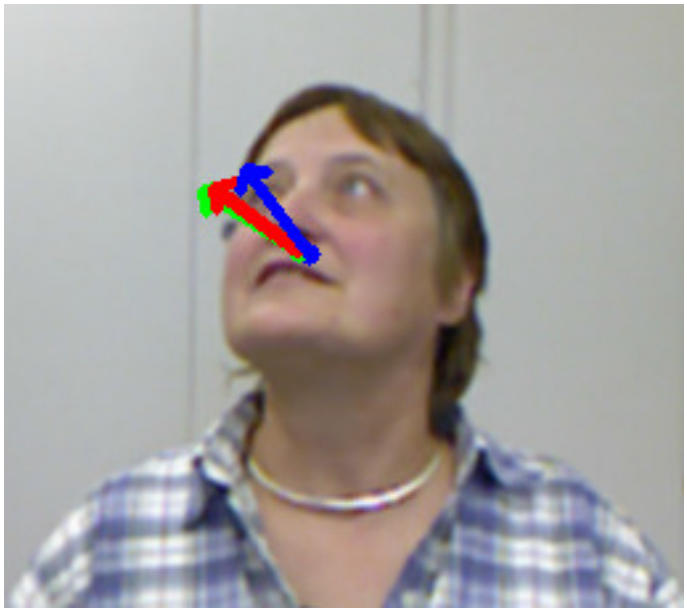


Figura 2: Representación de la orientación de la pose: flecha verde (estimación), flecha roja (estimación tras el proceso de tracking) y flecha azul (orientación real).

A continuación procedemos a analizar, de forma global y resumida, el funcionamiento del estimador de pose. Su herramienta principal son los Hough Forests [9, 1], que son un conjunto de árboles de decisión que pueden traducir espacios de entrada complejos en espacios de salida más simples, discretos o continuos. Los HF son considerados como un conjunto de clasificadores, los árboles, que permiten realizar una clasificación o regresión, que necesita ser consolidada en una fase final. Para el problema concreto que nos atañe, y así poder realizar su entrenamiento, partimos de un conjunto de muestras o parches anotados $P = \{P_i\}$.

En el caso de la estimación de pose de la cabeza los vectores estimación contienen la siguiente información: $\theta^1 = (\theta_x, \theta_y, \theta_z)$, para localizar el centro de la nariz de la cabeza a estimar, y $\theta^2 = (\theta_{dirección}, \theta_{elevación}, \theta_{alabeo})$ para estimar la orientación de la pose de la cabeza. En cambio, en el caso de la estimación de pose de vehículos en imágenes de tráfico, θ^1 tan sólo contiene información de θ_x y θ_y, para localizar el centro de cada vehículo, y en términos de orientación de la pose de los vehículos, el vector θ^2 se compone de los ángulos de acimut y zenit, para reflejar la trayectoria que sigue cada uno de los vehículos a estimar. Los parches P_i, son extraídos de forma aleatoria de las imágenes de entrenamiento, y forman las hojas de cada árbol T_t, y por lo tanto del bosque $F = \{T_t\}$. Cada árbol se encarga de convertir el problema original en problemas más pequeños, a través de todos sus nodos que no sean nodo hoja, aplicando un test binario en cada uno de dichos nodos, encaminando así la muestra introducida por el árbol, al nodo hijo izquierdo o al derecho dependiendo del resultado del test binario, de tal modo que la entropía de los conjuntos que se forman se vaya minimizando. Finalmente el valor más *votado* por todas las hojas, será reconocido como el resultado final de la estimación realizada por el bosque.

Cabe destacar que el presente trabajo de investigación no se ha encargado del entrenamiento los HF (aspecto que se ha resuelto en el marco de un proyecto de investigación dentro del grupo de trabajo). Este Trabajo de Fin de Máster se ha centrado en la implementación de la fase de test o de predicción, empleando mecanismos de tracking y de Kernel Density Estimation (KDE) para aumentar la precisión de las estimaciones, así como aumentando el campo de aplicación de uso del sistema de un demostrador de pose en rostros a un estimador también en imágenes de tráfico, capaz de clasificar los puntos de vista de los vehículos que aparecen en la escena.

En la Figura 3, se muestra el flujo de ejecución que lleva a cabo el sistema para la realización de las estimaciones deseadas. Las imágenes de entrada al estimador pueden ser de cabezas humanas o de tráfico. Posteriormente, se realiza el proceso de estimación donde, en primer lugar se extraen pequeños fragmentos o parches de la imagen a estimar para ser lanzados a través de los árboles. Posteriormente las hojas realizan su votación por un centro y orientación determinados, para seleccionar finalmente aquellos valores con un mayor cúmulo de votaciones a partir del resultado que proporciona el módulo de KDE cuyo cometido es estimar la función de densidad de probabilidad de una variable caracterizada por algún grado de aleatoriedad.

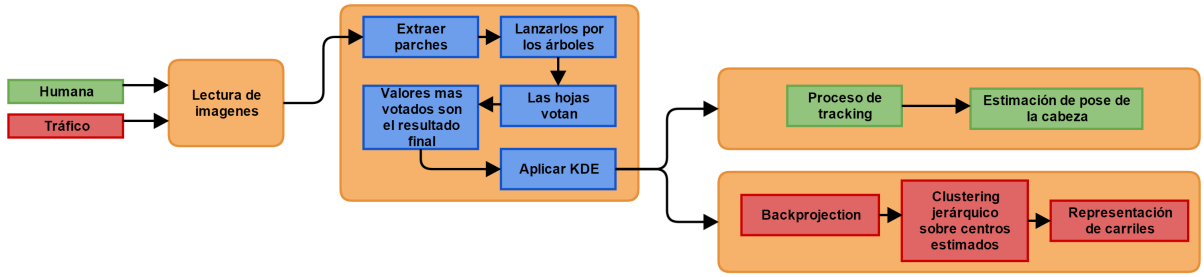


Figura 3: Ejecución del sistema en función de si el análisis es relativo a pose en cabezas o en vehículos.

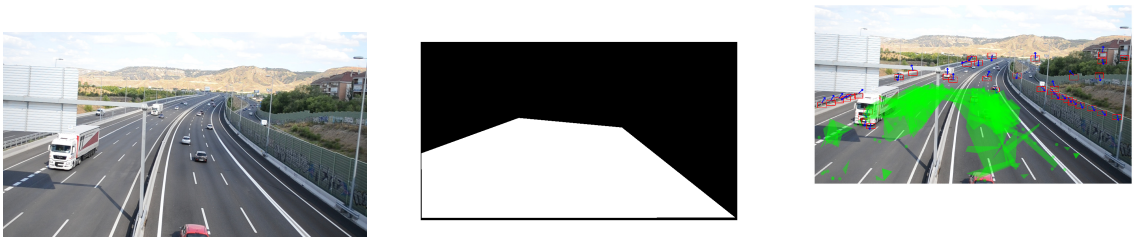
Finalmente, presentamos los principales resultados obtenidos en ambos problemas. A continuación se podrá ver reflejado el funcionamiento del estimador de pose de cabezas, en una secuencia de imágenes relativamente consecutivas, que representan el movimiento de una cabeza como podría producirse en cualquier caso de uso real. Las 3 representaciones de la pose anteriormente comentadas se pueden observar en la Figura 4, donde se ve como la flecha roja (tracking) se encuentra en casi todo momento entre la estimación (de color verde) y la orientación real (flecha azul), lo cual denota una mejora de precisión con respecto a no usar el bloque de tracking introducido en este trabajo.

Figura 4: Secuencia de imágenes de cabeza con las estimaciones de pose realizadas por el sistema.

Por otro lado, a continuación se muestra el funcionamiento del sistema de estimación

de pose de vehículos, para imágenes de tráfico, donde, tras la realización de la estimación de centros y respectivas orientaciones sobre dicha imagen, se aplica una máscara de imagen para eliminar de la misma las zonas exteriores que no corresponden con la calzada. Posteriormente, a través de clustering jerárquico y la acumulación de estimaciones en fotogramas no consecutivos se consigue construir la forma de los carriles que componen la imagen, principalmente utilizando la información de la pose de los vehículos, ya que es esta la que nos indica el sentido de circulación de los mismos.

En la Figura 5 se puede observar el proceso de estimación desde un fotograma inicial de una imagen de tráfico 5(a), procedente de una cámara de videovigilancia. Posteriormente, se aplica una máscara de imagen 5(b), para realizar el filtrado de aquellas estimaciones erróneas que se puedan producir en las zonas externas a la calzada. Por último, se aplica la técnica de clustering jerárquico acumulativo sobre la imagen estimada 5(c), donde se realizan agrupaciones de estimaciones de pose de todas las detecciones realizadas a lo largo del tiempo, sin que éstas estén relacionadas de forma directa, como pudiera tratarse de un vídeo donde un fotograma y el siguiente tuvieran relación. Por tanto, se ve finalmente construida una diferenciación de carriles que permita estimar la dirección del flujo de tráfico.



(a) **(b)** **(c)**

Figura 5: Imagen de tráfico incluyendo representación de carriles.

Glosario

- **RGB** Red/Green/Blue
- **Parches** Agrupaciones de píxeles
- **HF** Hough Forest
- **HRF** Hough Random Forest
- **RF** Random Forest
- **KF** Kalman Filter
- **PF** Particle Filter
- **DGT** Dirección General de Tráfico
- **NMS** Non-Maximum Supression
- **3D** 3 Dimensiones
- **Suavizado** Proceso de aproximación de valores, cuyo objetivo establece la eliminación de ruido en los mismos.

Capítulo 1

Introducción

El objetivo de este proyecto, en términos genéricos, consiste en la implementación de un sistema de estimación de pose. Dicho sistema se va a centrar, en primer lugar, en estimar la pose en imágenes donde se encuentre una cabeza con el rostro frente a la cámara. Para ello, se ha optimizado el sistema de detección diseñado mediante la introducción en el mismo, de un sistema de tracking que permita evitar grandes diferencias en cuanto al valor de la pose, en fotogramas próximos en el tiempo, con lo que se pueda ver reducido el error total de un proceso de estimación de forma notable. Por otro lado, el sistema se va a centrar en realizar estimaciones de pose sobre vehículos en imágenes de tráfico. El fin de lograr una correcta precisión en esta clase de estimaciones, radica en poder diferenciar claramente los carriles de circulación presentes en la imagen, mediante la agrupación de vehículos cuyos centros estén en posiciones cercanas, y con la misma orientación. Esta idea, puede ser introducida en mecanismos de videovigilancia existentes para aumentar la precisión a la hora de estimar flujos de tráfico y el estado de las carreteras de forma autónoma.

La clase de estimaciones que el sistema es capaz de realizar para ambos problemas se presentan en la Figura 1.1.

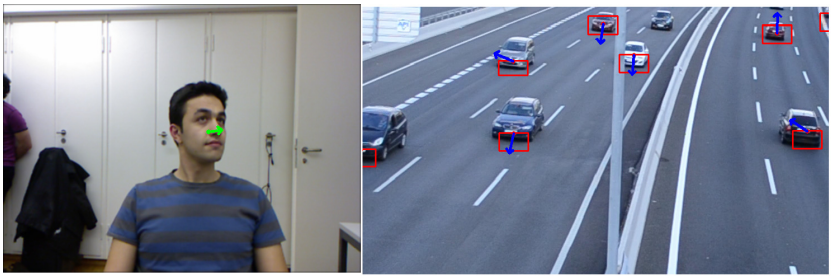


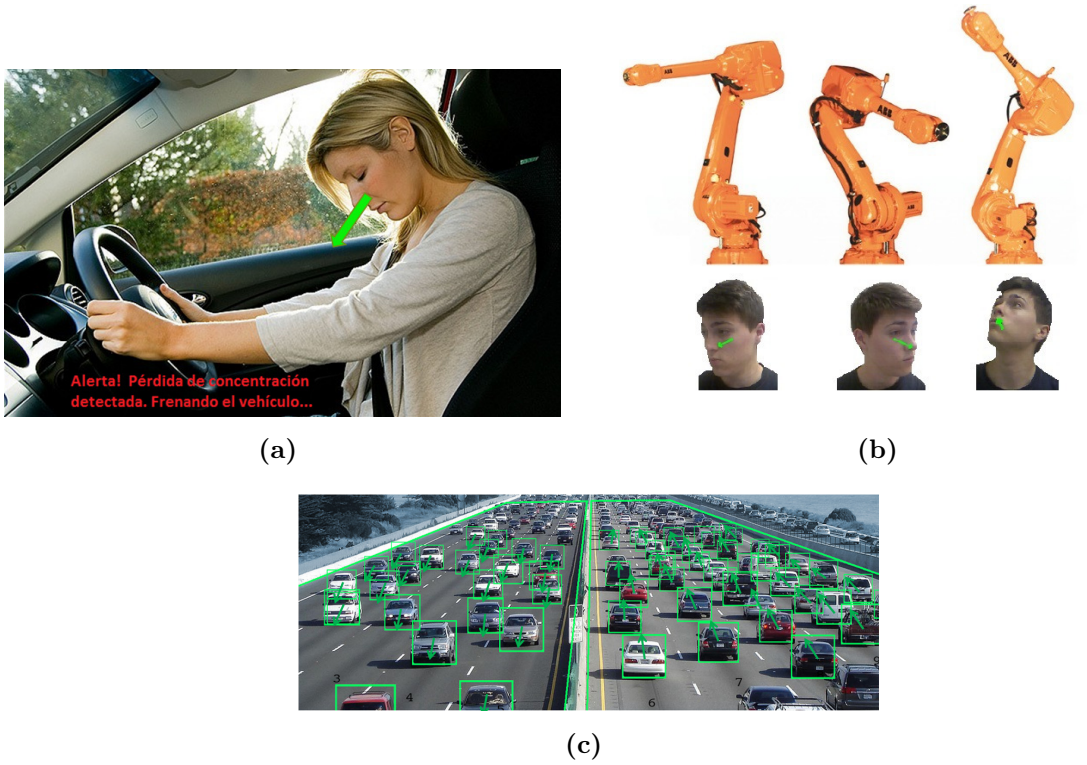
Figura 1.1: A la izquierda, se puede observar la estimación de pose en una persona que se encuentra frente a la cámara. A la derecha, se observan las detecciones de pose realizadas sobre vehículos en una imagen de tráfico.

1.1. Motivación y campos de aplicación

La principal motivación de este proyecto viene dotada de un carácter investigador en la temática de la estimación de pose a partir del tratamiento de imágenes por computador. A través de la implementación de las soluciones propuestas para la detección, seguimiento y estimación de pose de objetos, se pretenden precisar y agilizar tareas dentro de los campos de aplicación de la interacción humano-máquina y la videovigilancia en carretera. En concreto, este sistema dispone de una serie de campos de aplicación donde su implementación puede potenciar diversos aspectos en lo referente a:

- Sistemas de detección de fatiga en vehículos, a través del cual, se podrían evitar accidentes de tráfico a causa de despistes o fatiga excesiva en el conductor, mediante la detección de la pose de su cabeza para alertar cuando ésta no se encuentre orientada hacia la calzada.
- Sistemas de interacción humano máquina, para controlar toda clase de nuevas tecnologías, mediante un simple movimiento de cabeza. A su vez, también se puede emplear para el control de robótica en ámbitos de ocio e industria.
- Sistemas de conteo y video vigilancia a través de cámaras de tráfico en carretera, para agilizar el conocimiento acerca del estado de la calzada en un instante concreto, proporcionando una mayor precisión en ambas tareas ya que el filtrado realizado por los diferentes carriles estimados, no permite detecciones de vehículos que no se encuentren en dichos carriles.

A modo de ejemplo, se pueden observar en la Figura 1.2 las posibles aplicaciones del estimador de pose, que han sido comentadas anteriormente, comenzando por la detección de fatiga en vehículos, la interacción humano-máquina y por último la detección de pose en imágenes de tráfico para posibilitar una diferenciación en carriles.



(a) **(b)**

(c)

Figura 1.2: Empleo del sistema de estimación en campos de aplicación reales. a) Avisador de situación de despiste, por parte del conductor de un vehículo. b) Movimiento de robótica industrial pesada, con un simple movimiento de cabeza. c) Detección de retenciones repentinas y sus respectivas causas, mediante el conteo de vehículos teniendo en cuenta carriles de circulación separados.

Esta memoria de Trabajo de Fin de Máster va a comenzar introduciendo en el capítulo 2, el funcionamiento del estimador de pose, así como de los diferentes bloques para su optimización (tracking y KDE) de forma que el lector comprenda a la perfección el por qué se han incluido cada uno de dichos bloques en el sistema inicial de estimación. Una vez descrita la teoría en términos generales, se va a proceder a describir la ejecución del sistema dependiendo de la aplicación para la cual se destine, interacción con humanos (ver capítulo 2) o con vehículos (ver capítulo 3). Posteriormente, en el capítulo 4 de resultados, se va a proceder a mostrar los resultados obtenidos en términos de precisión y velocidad, en los dos campos de aplicación para los cuales dicho sistema se encuentra destinado. Esto se va a describir con el objetivo de que el lector conozca los aspectos mas innovadores que introduce el sistema a la línea de investigación, así como aquellos aspectos que podrían ser potenciados en un futuro en vías de investigación paralelas. Por último, para finalizar la redacción del presente libro, se van a extraer una serie de conclusiones del trabajo de investigación realizado, así como se van a referenciar una serie de futuras líneas de investigación que se podrían seguir a partir del presente sistema estimador.

1.2. Estado del arte

La literatura contiene varios trabajos en estimación de pose de cabeza [23], que pueden ser convenientemente divididos dependiendo de si se hace uso de imágenes RGB, datos de profundidad o ambos.

En general, las aproximaciones que hacen uso de imágenes RGB son sensitivas a la iluminación y a la falta de características distintivas. Por lo tanto, algunos de los trabajos recientes usan la profundidad como elemento primario. Breitenstein *et al.* [4], primero computa las hipótesis de posiciones de la nariz a partir de imágenes de profundidad de alta resolución, para después minimizar una función de error entre estas hipótesis y las imágenes de referencia de la pose. Posteriormente, algunos trabajos proponen el uso de un RF combinado con una estrategia de votación de Hough para solucionar el problema [9, 26, 11, 10]. Fanelli *et al.* [10] introducen el uso de RF para la estimación de pose de la cabeza en tiempo real a partir de escaneos de rango de alta calidad. La aproximación fue posteriormente adaptada a datos de profundidad, a partir de cámaras de profundidad comerciales [9, 11]. En [26], los autores proponen extender la votación en HF con el concepto de Probabilistic Locally Enhanced Voting (PLEV), una estrategia de regresión que consiste en modular la regresión con una estimación de densidad del núcleo, para consolidar los votos en una región local cerca del máximo detectado en el espacio de Hough. Schulter *et al.* [28] ha mejorado los RF proponiendo los nuevos Alternating Regression Forest, que son capaces de minimizar las pérdidas globales para obtener una mejor generalización, en contraste con el proceso de minimización local empleado durante el aprendizaje de RF tradicionales. Finalmente, Riegler *et al.* [27] describe un método que combina redes neuronales convolucionales con la idea de Hough Forests para una estimación continua de la pose de la cabeza.

Con los algoritmos basados en imágenes 2D, nos enfocamos aquí en métodos basados en apariencia. En contraste con nuestro modelo, una aproximación basada en apariencia común es discretizar las poses de la cabeza y aprender un detector separado para cada una de ellas, e.g. [20, 22, 8, 14]. Numerosos trabajos confían en modelos estadísticos de la forma de la cara y apariencia, e.g, Active Apperance Models [5] y sus extensiones [6, 25, 29], pero su enfoque se centra habitualmente en la detección y seguimiento de las características faciales. Estos métodos confían en cuantizaciones gruesas de las poses para detección de rostro en múltiples vistas, en lugar de considerar que la estimación de la pose es a la larga un problema continuo. Recientemente, Demirkus *et al.* [7] han propuesto un modelo gráfico temporal jerárquico para estimar un ángulo continuo de la pose de la cabeza a partir de vídeos reales. La metodología introducida proporciona una función de densidad de probabilidad (PDF) de las poses de la cabeza para cada fotograma de vídeo, en lugar de una decisión única. Sin embargo, solo el ángulo de dirección es considerado, mientras que nuestra solución realiza la estimación de forma simultánea para los ángulos de elevación, dirección y alabeo.

Nosotros construimos nuestra aproximación en el trabajo descrito en [26]. Por lo tanto, la nuestra es una estrategia basada en HF. Sin embargo, a diferencia de [28, 9, 11, 10], presentamos un modelo que no necesita información de profundidad sino imágenes RGB. Específicamente, hemos desarrollado una implementación compacta del HF+PLEV [26] haciendo uso de C++. Para disponer de un sistema rápido para HCI, hemos extendido el HF+PLEV [26] para trabajar con un filtro de preprocesado para la selección de las regiones de rostro candidatas, implementando una etapa de detección de color de piel. Este preprocesado acelera las estimaciones de pose. Finalmente, alimentamos un modelo de tracking únicamente con las estimaciones de pose de las regiones de cara candidatas para mejorar las predicciones. Nosotros evaluamos el uso de dos soluciones de tracking: un filtro de Kalman [33] y un algoritmo de Condensación [19]. Todas estas extensiones nos permiten construir un modelo final que es más rápido que el original HF+PLEV [26], así como también más preciso – nuestros resultados muestran que nuestra implementación mejora los resultados reportados en [26].

En lo referente a estudios basados en la estimación de carriles, no se ha encontrado un trabajo de investigación que afronte la problemática a través de la estimación de centros en vehículos, tal y como se realiza en este proyecto, sino que en algunos casos se aprovechan aquellas zonas de calzada que, por el continuo paso de vehículos, ven su característica de color continuamente variada con lo que se puede identificar como una porción de carril, pudiendo formar cada carril completo a través de la suma de todos estos fragmentos [24]. En otros casos en cambio, se centra el estudio en los rasgos más característicos de la propia calzada [18], como son las pinturas presentes en la carretera y las curvas que los propios carriles forman, para determinar qué zonas de la imagen analizada forman parte de un carril. En nuestro caso concreto, adicionalmente a la detección de las zonas de la imagen que forman un carril de carretera, se proporciona información referente al sentido hacia el cual fluye el tráfico en dicho carril, gracias a la detección de pose de los vehículos que atraviesan dicho carril.

Capítulo 2

Detección y estimación de la pose de rostros humanos

Este capítulo procede a describir el modelo teórico empleado para la implementación del estimador de pose de cabezas humanas, así como los resultados obtenidos tras los diferentes procesos de test y análisis realizados sobre el mismo.

2.1. Aprendiendo un HF para Estimación de Pose en Cabezas

Un Random Forest (RF) típico [3] es un clasificador consistente en un conjunto de árboles de decisión binarios aleatorizados. Durante el entrenamiento, un árbol de decisión binario débil es aprendido para cada nodo no hoja. Durante la ejecución, las muestras de test son lanzadas a través de los árboles, y la salida es computada a través del promedio de las distribuciones aprendidas en los nodos hoja. Los HF [13] son una generalización de la transformada de Hough dentro del marco de los RF. Los árboles aleatorios son entrenados para aprender un mapeado a partir de características d-dimensionales muestreadas para sus correspondientes votaciones en un espacio de Hough $\mathcal{H} \in \mathbb{R}^H$.

Para el problema propuesto, nos hemos basado en la aproximación de Redondo-Cabrera *et al.* [26]. Tal y como se describe en [26], en el HF $\mathcal{F}$, agregamos un conjunto de T árboles de decisión binarios $\mathcal{T}_t(P_i) : \mathcal{P} \rightarrow \mathcal{H}$, donde $\mathcal{P} \subset \mathbb{R}^d$ es el espacio de característica d-dimensional y $\mathcal{H} \subset \mathbb{R}^h$ describe el espacio de Hough donde las hipótesis son codificadas. Este espacio de Hough nos permite recuperar hipótesis para la localización y pose continua de la cabeza en múltiples escalas. Cada hipótesis de cabeza $\mathbf{h} \in \mathcal{H}$ puede ser definida como $\mathbf{h} = (x_h, y_h, \theta_h, s_h)$, donde x_h y y_h hacen referencia al centro del rostro (en nuestro caso, el centro de la nariz), $\theta_h \in \mathbb{R}^p$ representa la pose continua, y s_h identifica la escala a la que se detecta el objeto.

Cada uno de los árboles de decisión $\mathcal{T}$ se construye a partir de parches muestreados $P_y = \{(\mathcal{I}_i, c_i, d_i, \theta_i)\}$, donde $\mathcal{I} = \left\{I^1, I^2, ..., I^N\right\}$ es la apariencia de la imagen de entrena-

miento, I^j es el j^{th} canal de apariencia, $c_i \in \mathcal{C} : \{0, 1\}$ es una etiqueta de clasificación (0 para una muestra de fondo de la imagen y 1 para una muestra de cabeza), $d_i = (x_i, y_i)$ representa la localización relativa en 2D del centro de la cabeza al parche muestreado. Además, θ_i define la pose continua de la cabeza. El objetivo del proceso de entrenamiento, consiste en construir los árboles de decisión, de modo que los parches con información similar queden agrupados en las mismas hojas. El proceso de entrenamiento, para un árbol de un bosque aleatorio es el siguiente. El entrenamiento comienza considerando a todos los parches en un mismo nodo, el nodo raíz S_0. Para la realización de la primera división en dos grupos, el modelo necesita hacer uso de un clasificador, denominado test binario, el cuál separa los parches del conjunto de entrada S en dos subconjuntos S^D y S^I, con el objetivo de que la entropía de los subconjuntos sea mínima, calculada de acuerdo a una de las variables de entrenamiento disponibles, y que son elegidas en cada nodo de forma aleatoria. En este trabajo de investigación, un nodo puede ser optimizado para minimizar la entropía de clasificación utilizando la información de localización d, la información de pose θ o las etiquetas c de los parches. Por tanto, esto significa que si el árbol decide optimizar de acuerdo a la localización d, el clasificador binario se optimizará para que los parches con localizaciones d similares, sean agrupados en los grupos S^D y S^I. En resumen, entrenar un árbol de decisión implica la división de forma recursiva de cada nodo para que los datos de entrenamiento en nuevos nodos hijos sean puros de acuerdo con la etiqueta de clase, la localización 2D relativa y la pose. Cada árbol crece hasta que sucede un criterio de parada, *p.ej.* fijando un máximo de profundidad de árbol o un número mínimo de parches muestreados.

En nuestro caso, al igual que en [26], la función de división $f(\mathcal{I}_i; \tau_1, \tau_2, \{R_r\}_{r=1}^4)$ se encuentra caracterizada por los siguientes parámetros: el canal de apariencia especificado por $\tau_1 \in \{1, 2, ..., N\}$, cuatro rectángulos asimétricos definidos mediante el parche $\{R_i\}_{i=1}^4$, y un umbral $\tau_2 \in \mathbb{R}$ para la diferencia en la media de valores de las áreas rectangulares. Nosotros entonces definimos la función de división por nodo como sigue:

$$f(\mathcal{I}_i; \tau_1, \tau_2, \{R_r\}_{r=1}^4) = \begin{cases} 0 \text{ si } f_a(\mathcal{I}_i; \tau_1, \{R_r\}_{r=1}^4) < \tau_2, \\ 1 \text{ en caso contrario.} \end{cases} \tag{2.1}$$

con $f_a(\mathcal{I}_i; \tau_1, \{R_r\}_{r=1}^4) = |R_1|^{-1} \sum_{q \in R_1} I_i^{\tau_1}(q) - \sum_{r=2}^{4} \left(|R_r|^{-1} \sum_{q \in R_r} I_i^{\tau_1}(q) \right)$. Esta ecuación consiste en la inclusión de cuatro rectángulos asimétricos por cada parche extraído de la imagen. La operación se fundamenta en la resta de, la suma de los valores de uno de los 32 canales de características (el canal denominado τ_1) computada en la región R_1, y la suma de los valores del mismo canal en otras 3 regiones. Se debe notar el hecho de que la posición de las regiones rectangulares son obtenidas de forma aleatoria.

Cada nodo elige la mejor función de división a través de la optimización de una de las 3 siguientes medidas de impureza, $\mathcal{M}_*(\mathcal{S})$, que son elegidas de forma aleatoria durante el entrenamiento. La impureza de la etiqueta de clasificación se mide como en [1] mediante

$$\mathcal{M}_c(S) = H(S) - \sum_{hijo\in(l,r)} \frac{S^{hijo}}{S} H(S^{hijo}), \tag{2.2}$$

dónde $H(S)$ es la entropía proporcionada por $H(S) = -\sum_{c=0}^{1} p(c|S)\log(p(c|S))$, y $p(c|S)$ indica la distribución empírica sobre las clasificaciones dentro del conjunto S.

La impureza de la localización 2D relativa del parche, como en [13], es definida por

$$\mathcal{M}_d(S) = \sum_{hijo\in(l,r)} \sum_{j:c_j=1} ||d_j - \frac{1}{|S^{hijo}|} \sum_{k:c_k=1} d_k||^2. \tag{2.3}$$

Esta ecuación, se basa en operar con la diferencia entre la información de localización presente en una etiqueta d_j con respecto a la información media de clasificación d_k en los parches de la zona facial, donde la etiqueta de clasificación tiene un valor de $c_k = 1$, todo ello computado en el conjunto S^{hijo}.

Y la impureza de la pose de la cabeza, mediante la cual es conocida la incertidumbre de acuerdo a la orientación de cada etiqueta de clasificación, es computada como en [26]:

$$\mathcal{M}_p(S) = \sum_{hijo\in(l,r)} \sum_{j:c_j=1} \left(\frac{min\{(||\theta_j - \theta_A||), 360° - (||\theta_j - \theta_A||)\}}{180°} \right)^2, \tag{2.4}$$

dónde se opera con la diferencia entre la información relativa a la orientación presente en una etiqueta, θ_j menos θ_A, que es la media del ángulo de orientación sobre todos los parches correspondientes a zona de cabeza en el conjunto S^{hijo} y es computada teniendo en cuenta el ciclo natural de los ángulos de pose. Este cálculo matemático de la orientación media obtenida, proviene del concepto de *mediadecantidadescirculares*, y más concretamente del cálculo de ángulos medios.

Finalmente, se presentan las probabilidades de clasificación $p(c = k|P)$ y las distribuciones continuas de los parámetros de la pose de la cabeza: $p(d = N(d; \overline{d}, \sum^d)$ y $p(\theta = N(\theta; \overline{\theta}, \sum^\theta)$

2.2. Detección Rápida de Cabeza y Estimación de Pose para HCI

La interacción humano-máquina o Human-Computer Interaction (HCI) está cada vez atrayendo mayor atención. Dado que la gente interactúa a través de medios que emplean diferentes canales, incluyendo postura corporal y pose de cabeza, un paso importante a lo largo de interfaces mas naturales es el análisis visual de los movimientos del usuario empleando una máquina. A través de la interpretación de los movimientos de todo el cuerpo, como se ha realizado para sistemas como el destinado al ocio Kinect, nuevos

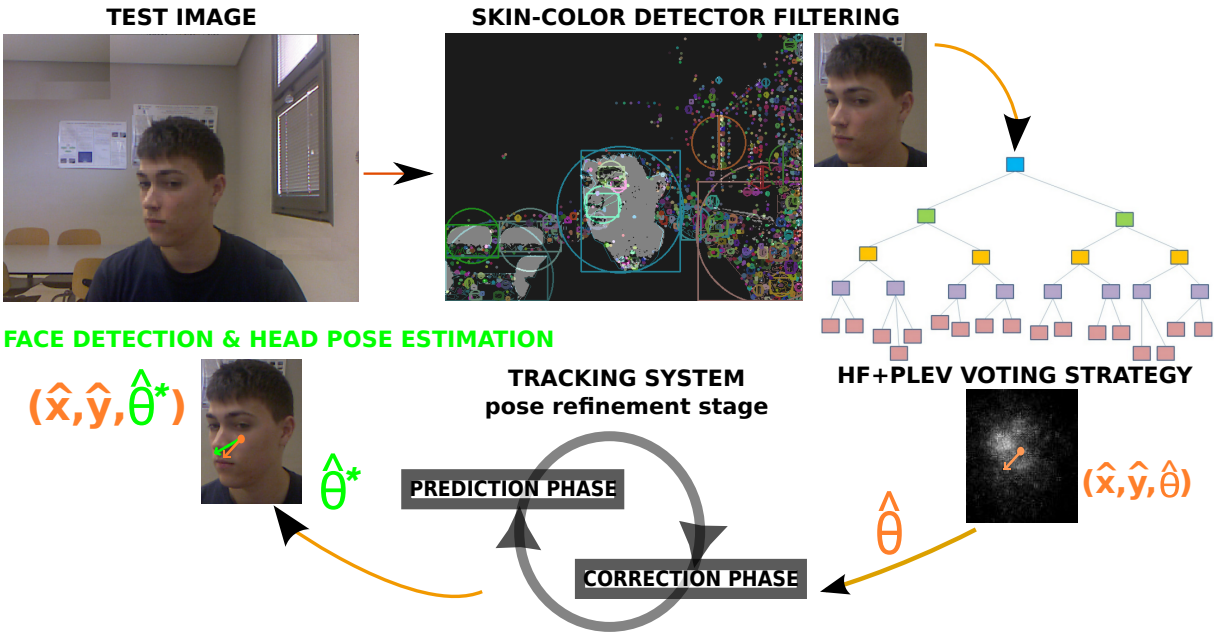


Figura 2.1: Nuestra aproximación es capaz de estimar de forma conjunta la localización y la pose continua de la cabeza. Comenzamos realizando un filtrado de color de piel, y entonces seguimos una votación de regresión HF + PLEV [26], en conjunto con un paso de tracking para consolidar las estimaciones finales de la orientación de la cabeza.

interfaces beneficiarían a través de la estimación rápida de la pose de la cabeza, como el que se presenta en este Trabajo de Fin de Máster.

El actual estado del arte en métodos basados en el principio de los Hough Forests (HFs) [13] ha demostrado ser muy exitoso en la detección y estimación de pose de cabeza (*p.ej* [9, 26, 11, 10]). Todos estos métodos entrenan un Random Forest (RF) [3] para un proceso conjunto de clasificación y regresión. Los parches son muestreados de forma densa desde la imagen y el modelo determina para cada parche si éste pertenece a una zona del rostro o si pertenece a la zona de fondo de la imagen. Adicionalmente, para parches de cabeza, el modelo realiza una regresión para localizar el centro de la cabeza y estimar su pose. Sin embargo, mientras los métodos presentados en [9, 11, 10] requieren imágenes de profundidad, el HF con la Votación Probabilística Localmente Mejorada (PLEV) descrito en [26] ha alcanzado los resultados del estado del arte haciendo uso de imágenes RGB únicamente.

Aquí, los miembros del equipo de investigación GRAM se han encargado de incluir el modelo PLEV en la estimación a través de HF, así como de diseñar una aproximación para disponer de una estimación de pose rápida (ver Figura 2.1)

Nuestra solución es capaz de realizar la detección y estimación haciendo uso de una simple cámara web. Al sistema se le ha incorporado un filtrado de color de piel y una etapa de tracking para refinar las estimaciones de pose. La implementación ha sido desarrollada completamente en C++, que produce que nuestro modelo sea notablemente mas rápido

que la aproximación original de [26]. Además, la validación experimental muestra que la solución propuesta es capaz de superar los resultados de estado del arte en la *Base de Datos de Pose de Cabeza Kinect Biwi* (Biwi) [9] haciendo uso de imágenes RGB únicamente.

Dada una imagen de test, ver Figura 2.1, comenzamos realizando un método basado en la detección de píxeles de piel. Esto puede ser considerado como un paso de preprocesado para encontrar las regiones de rostro del candidato.

Nuestro detector de píxeles de piel emplea un método de umbralización para la clasificación de píxeles de piel y no piel [31]. Esta clase de detector de piel define los límites de la zona de piel en ciertos espacios de color haciendo uso de un conjunto de umbrales fijos de color de piel. Los píxeles que corresponden a zonas de piel son blancos, y los que pertenecen al trasfondo, figuran como negros. Para comenzar, se realiza un filtrado Gaussiano Blur de la imagen original, para suavizar los altos contrastes de color en la imagen, viéndose disminuido el ruido en la misma. Tras ello, se continua con la conversión de valores RGB de cada píxel de la imagen, a valores YCbCr, ya que esta escala de color permite mejorar el rendimiento de la segmentación de piel ante cambios en el brillo de la imagen, ya que sus valores se ajustan precisamente en términos de luminancia y crominancia. Dichos valores se han calculado a partir de los estándares que definen los rangos de color entre los cuales se encuentra con mayor probabilidad los píxeles de la piel humana. El proceso de realización de contornos comienza fijando un umbral, que como se ha comentado, ayudará a distinguir píxeles blancos de píxeles negros, en la imagen procesada por la máscara de piel. A continuación se muestra la conversión de espacio de color para pasar las características RGB de la imagen de entrada a YCbCr.

$$\begin{bmatrix} Y \\ C_b \\ C_r \end{bmatrix} = \begin{bmatrix} 0{,}299 & 0{,}587 & 0{,}114 \\ -0{,}169 & -0{,}331 & 0{,}500 \\ 0{,}500 & -0{,}419 & -0{,}081 \end{bmatrix} \bullet \begin{bmatrix} R \\ G \\ B \end{bmatrix} + \begin{bmatrix} 0 \\ 128 \\ 128 \end{bmatrix} \tag{2.5}$$

Para la realización del cálculo se debe tener en cuenta que; $Y \in [0, 255]$, $C_b \in [0, 255]$, $C_r \in [0, 255]$. Tras disponer de la ecuación, se proceden a mostrar los valores fijados a modo de umbral para la segmentación de piel. Este umbral ha sido seleccionado tras la realización de un proceso de ajuste, siendo el que mejor funcionamiento tiene en relación a la asignación de píxeles blancos para píxeles que realmente se corresponden con la piel, teniendo en cuenta las posibles variaciones del brillo en la imagen.

$$ColorPixel = \begin{cases} Blanco & \text{si } 15 < Y < 235, 60 < C_b < 122, 135 < C_r < 170, \\ Negro & \text{en caso contrario.} \end{cases} \tag{2.6}$$

Tras disponer de una diferenciación entre píxeles blancos y negros, se buscan los contornos que encierren un área de píxeles blancos, y de manera gráfica, se aproxima la forma de cada contorno a la de un polígono, para que estas áreas puedan ser delimitadas de forma simple en un rectángulo o círculo, que indiquen la zona que se debe extraer de la imagen original para ser estimada.

Tras el proceso de búsqueda de contornos en la imagen, se realiza un filtrado adicional, dado que en ocasiones la zona de fondo de la imagen no solo contiene píxeles negros, sino que debido al brillo algunos píxeles pueden ser detectados como blancos. Este filtrado consiste en descartar aquellos contornos que no cumplen unas reglas de tamaño y proporción, las cuales han sido definidas mediante un proceso de ajuste teniendo en cuenta unos límites máximos de lejanía de la cabeza con respecto a la cámara, así como la proporción de dicha cabeza en la imagen, evitando así contornos con proporciones muy desiguales, que en casi la totalidad de los casos no se van a corresponder con la zona de la cara humana.

$$Dimensiones\ del\ contorno = \begin{cases} Pasa\ el\ filtro & \text{si } 4{,}000 < pixeles < 40{,}000, \\ Descartado & \text{en caso contrario.} \end{cases} \tag{2.7}$$

En términos de proporción, los límites impuestos son los siguientes:

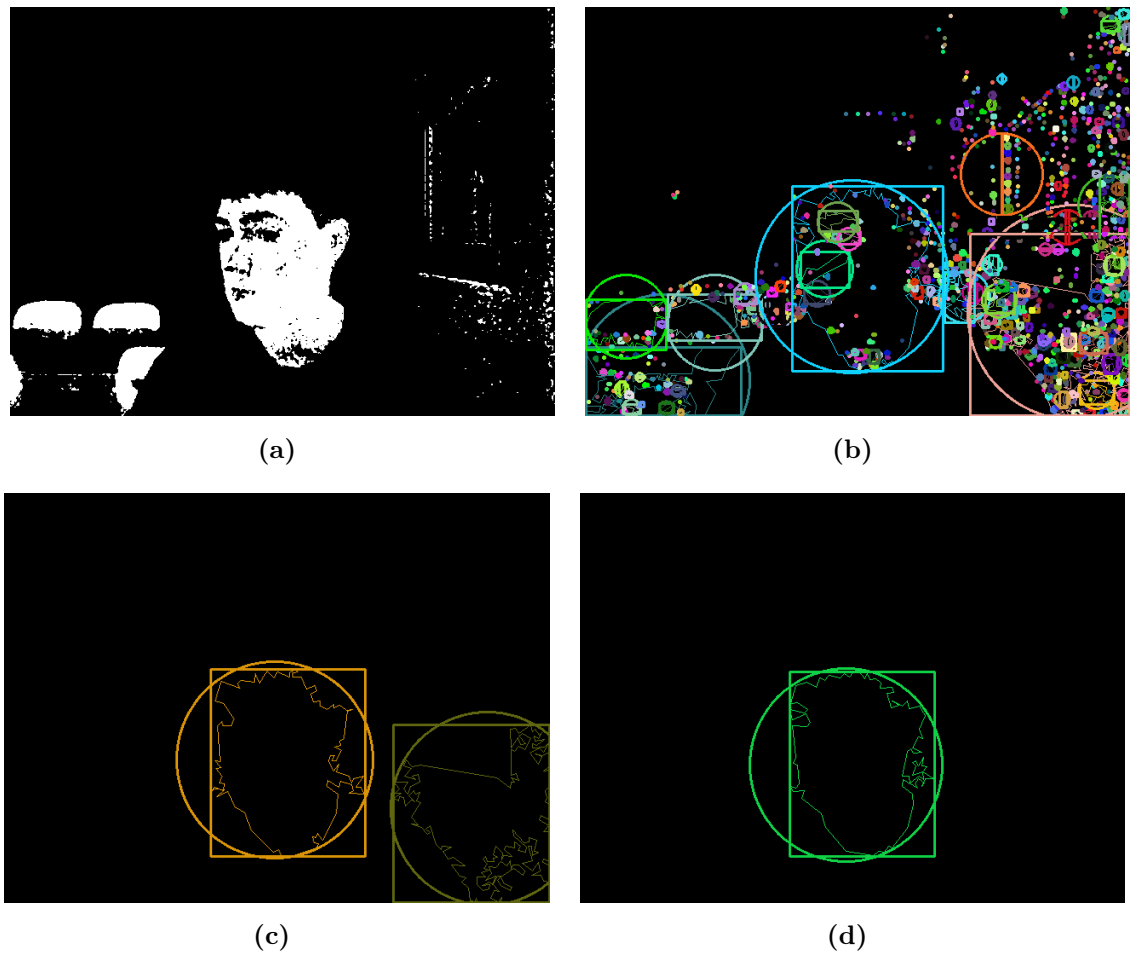
$$Proporción\ del\ contorno = \begin{cases} Pasa\ el\ filtro & \text{si } 0{,}85 < \frac{alto}{ancho} < 2, \\ Descartado & \text{en caso contrario.} \end{cases} \tag{2.8}$$

En caso de que sólo uno de los contornos inicialmente considerados como válidos, por el proceso de segmentación, pase todos los filtrados, será éste el que se corresponda con la cabeza a estimar. En cambio, si mas de un contorno sobrepasa todos los filtros expuestos, se debe proceder a realizar un último filtrado que escoja únicamente un sólo contorno. Este filtrado, consiste en recuperar el valor de la puntuación que caracteriza la última detección realizada, y si ésta es superior a un valor de 0.10, el contorno que resultará seleccionado, será el más próximo al último centro de la cabeza estimado. El sentido de este proceso de filtrado radica en que, si el valor de la puntuación dada a la última estimación es inferior a 0.10, esto querrá decir que difícilmente dicha detección se corresponderá con una zona de interés, y por tanto, en tal caso se procederá a escoger el contorno situado más próximo al centro de la imagen, que a su vez, no sea el más cercano al último centro detectado. Esto se realiza con el objetivo de no permanecer escogiendo continuamente un mismo contorno, que no se corresponde con un rostro humano.

En la siguiente Figura 2.2 se puede ver el proceso de actuación del detector de piel, donde intervienen a su vez una serie de umbrales de tamaño y proporción de las regiones de piel detectadas para producir la única detección de la zona de rostro en la imagen procesada.

Después del paso del detector de piel, los parches de la imagen son muestreados únicamente desde las regiones identificadas como posibles candidatas. También se ha incorporado un parámetro de salto de píxeles, que controla como de densa es la extracción de parches, en función de si se desea una mayor velocidad en la obtención de la estimación o una mayor precisión de la regresión.

Estos parches atraviesan los árboles aprendidos y traspasan las votaciones al espacio multidimensional de Hough $\mathcal{H} \subset \mathbb{R}^{2+p}$ basándose en las distribuciones de localización



(a) **(b)**

(c) **(d)**

Figura 2.2: (a) Imagen de salida del detector de piel, para los umbrales del espacio de color YCrCb fijados. (b) Acotación de las áreas reconocidas como piel en la imagen. (c) Aplicación de los umbrales de tamaño y proporcionalidad de las cajas detectadas. (d) Selección de la caja mas adecuada atendiendo a la que se encuentra en una posición mas céntrica.

y pose almacenados en las hojas. Notar que p es el número de ángulos que definen la pose continua de la cabeza. La estimación basada en bosques es entonces computada mediante la agregación de votaciones procedentes de diferentes parches a diferentes escalas $\{s_1, s_2, \ldots, s_S\}$. Siguiendo un estándar de aproximación a la regresión en HF [13], las votaciones son acumuladas de una forma aditiva en los correspondientes espacios de votación de Hough $\{\mathcal{H}^1, \mathcal{H}^2, \ldots, \mathcal{H}^S\}$, donde $\mathcal{H}^i \in \mathbb{R}^{2+p}$. Entonces, estos espacios de Hough son agrupados y escalados, con el objetivo de que el valor máximo de votación puede ser localizado de forma conjunta a múltiples escalas. Para localizar este valor máximo de votaciones, usamos el procedimiento descrito en [26] denominado PLEV, donde una región local de Hough ($H_r^{\hat{\mathbf{h}}} \subset \mathcal{H}^i$), en lugar de un único máximo de Hough, es considerada para la regresión de la pose. PLEV añade todas las votaciones de pose recibidas en $H_r^{\hat{\mathbf{h}}}$, obteniendo una distribución global $g_r^{\hat{\mathbf{h}}}$ en la región de Hough, que puede ser computada como,

$$g_r^{\hat{\mathbf{h}}} = \sum_{v_i \in H_r^{\hat{\mathbf{h}}}} \left(\sum_{L_j \rightarrow v_i} \frac{p(c{=}1|L_j)}{|L_j|} p(\theta|L_j, v_i) \right) , \tag{2.9}$$

dónde $p(c = 1|L_j)$ y $|L_j|$ codifican la probabilidad de zona de cabeza y el número de parches en la hoja L_j, respectivamente, $p(\theta|L_j, v_i)$ es la distribución de poses asociada a los parches en la hoja L_j que comparte una votación en $H_r^{\hat{\mathbf{h}}}$, que denotamos como $L_j \rightarrow v_i$.

Finalmente se realiza un KDE Gaussiano en la distribución $g_r^{\hat{\mathbf{h}}}$ para obtener una función de densidad de probabilidad mas suavizada para la estimación de pose,

$$f_{g_r^{\hat{\mathbf{h}}}}(\theta) = \frac{1}{|g_r^{\hat{\mathbf{h}}}|} \sum_{\forall g_r^{\hat{\mathbf{h}}}(i)} \frac{1}{\sqrt{2\pi}} \exp\left(\frac{(\theta - g_r^{\hat{\mathbf{h}}}(i))^2}{2h^2}\right) , \tag{2.10}$$

con h el ancho de banda y $|g_r^{\hat{\mathbf{h}}}|$ el número total de posiciones de votación consideradas. Finalmente, nuestra hipótesis final para la pose $\hat{\theta}$ es obtenida como $\arg\max_{\hat{\theta}} = f_{g_r^{\hat{\mathbf{h}}}}(\theta)$.

2.3. Tracking de la Estimación de Pose de Cabeza

La aproximación en [26] trabaja sobre cada fotograma de forma independiente. En este trabajo proponemos el refinamiento de la estimación de pose con una solución de tracking (ver etapa de refinamiento en la Figura 2.1). El objetivo es mejorar la precisión de las predicciones para los valores $\hat{\theta}$ de estimación de pose, evitando resultados ruidosos. Para realizar esto, se ha integrado en el sistema un bloque de tracking para obtener estimaciones refinadas $\hat{\theta}^*$. En nuestro modelo, el tracking es realizado para cada uno de los ángulos de la pose de la cabeza (dirección, elevación y alabeo) de forma independiente. Experimentalmente, se ha validado el uso de dos modelos de implementación de tracking: un filtro de Kalman (KF) [33] y un filtro de Partículas (PF) [19].

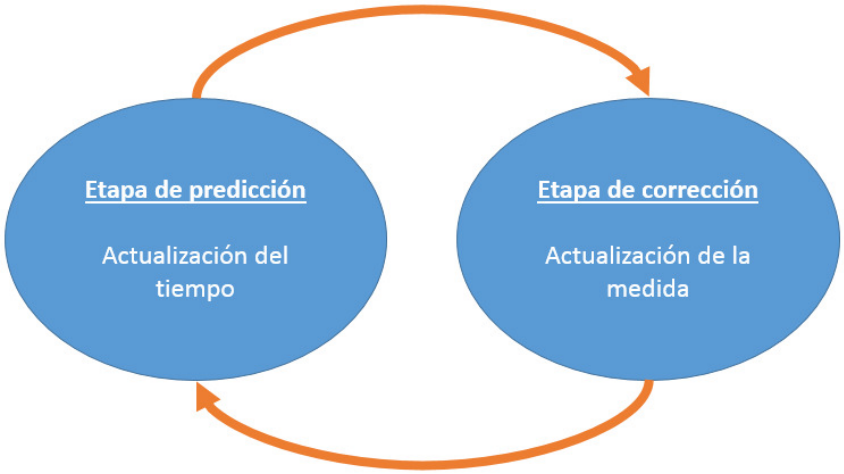


Figura 2.3: Ciclo de funcionamiento del filtro de Kalman.

2.3.1. Filtro de Kalman: Modelo e Implementación

El filtro de Kalman es un algoritmo empleado para la estimación de un proceso mediante el uso de lo conocido como, *control por realimentacin*. Este control es empleado por el filtro cuando este ha estimado un determinado estado del proceso, con el objetivo de obtener unas medidas procedentes de estados anteriores para así reducir el ruido blanco aditivo. Las ecuaciones que componen el filtro de Kalman son de 2 tipos: actualización del tiempo y actualización de la medida. La actualización del tiempo consiste en proporcionar a estados futuros del proceso, el estado actual, así como la medida de la estimación del error de covarianza, para que dicho estado futuro disponga de un valor denominado *a priori* desde el cual comenzar a basar su estimación en un estado concreto. Haciendo uso de esta estimación *a priori*, se pueden conseguir mejores resultados *a posteriori*, los cuales hacen referencia a los valores de salida estimados por el filtro de Kalman, en un estado concreto. Dada esta definición de la actualización del tiempo, se dice que esta ecuación es de tipo predictivo. Por otro lado, la actualización de la medida es precisamente la encargada de incluir la nueva medida, procedente de estados anteriores, en la estimación *a priori* del estado actual. Con ello se consigue cerrar el proceso de retroalimentación que comienza a partir de la actualización del tiempo. Por ello, esta ecuación es de tipo corrector, ya que tras esta ecuación se corrige el valor *a priori* proporcionado por la ecuación predictiva, para proporcionar un valor *a posteriori* lo mas ajustado a la realidad como sea posible. En la Figura 2.3 se puede observar la representación del ciclo básico de funcionamiento del filtro de Kalman, anteriormente descrito.

A continuación, se proceden a describir las ecuaciones de las diferentes etapas de funcionamiento del filtro de Kalman, comenzando por la etapa de predicción.

$$\hat{x}_k^- = A\hat{x}_{k-1} + Bu_{k-1} \tag{2.11}$$

$$P_k^- = AP_{k-1}A^T + Q \tag{2.12}$$

La primera ecuación hace referencia a la estimación del estado *a priori*. Para ello obtiene el valor de $\hat{x}_k^-$, que es la estimación *a priori* del vector de estados, a partir de la relación de A, que es la matriz de transición de estados con x_{k-1}, que es la estimación *a priori* del estado anterior $k-1$. Además, se añade la presencia de la matriz B, la cual es opcional, y se encarga del ajuste de las medidas realizadas por el filtro en un estado anterior $k-1$. En segundo lugar, se tiene la ecuación de cálculo de la covarianza del error asociada a la estimación *a priori*, la cual es obtenida a partir de la relación de la matriz de transición de estados A, con la covarianza del error calculada en el estado anterior $k-1$, añadiendo tras este cálculo, la presencia de la matriz de covarianza de ruido del proceso Q. Cabe destacar que esta matriz podría sufrir cambios a lo largo de la ejecución del proceso en determinadas aplicaciones, pero en el modelo de este trabajo se ha decidido hacer uso de unos valores fijos para caracterizar esta matriz.

En cuanto a las ecuaciones de actualización de la medida, se dispone de las siguientes:

$$K_k = P_k^- H^T \left(HP_k^- H^T + R\right)^- 1 \tag{2.13}$$

$$\hat{x}_k = \hat{x}_k^- + K_k \left(z_k - H\hat{x}_k^-\right) \tag{2.14}$$

$$P_k = \left(I - K_k H\right) P_k^- \tag{2.15}$$

La primera ecuación se emplea para obtener el valor de la ganancia de Kalman. Para ello se relaciona la covarianza del error de la estimación *a priori* con la matriz H, que hace referencia al estado de la medida realizada por el filtro en el instante actual z_k, además de añadir la matriz R que mide la covarianza del ruido en la medida realizada en dicho instante. En segundo lugar, se dispone de la obtención del estado *a posteriori*, el cual es dependiente del estado *a priori* x_k^- al cual se le suma el cálculo de la medida total en el estado actual, que depende de la ganancia de Kalman calculada en la anterior ecuación, multiplicada por la diferencia entre la medida realizada en el instante actual z_k menos la matriz H multiplicando al estado de la estimación *a priori*. Por último se realiza el cálculo de la estimación de la covarianza del error *a posteriori*, para conocer en que grado se ha desviado la estimación proporcionada por el filtro del valor real que debería tener.

En cuanto a la incorporación del filtro de Kalman en el sistema de estimación de pose de la cabeza, se ha empleado para reducir el ruido en la estimación de la orientación entre fotogramas consecutivos, evitando así grandes diferencias de orientación de la pose cuando difícilmente se podría dar el caso dada la proximidad temporal de los fotogramas analizados. Se ha incluido un filtro por cada ángulo que forma la orientación, haciendo un total de 3 filtros de Kalman, donde cada uno se encuentra caracterizado por unas matrices con diferentes valores, obtenidos mediante un proceso de ajuste que se comentará

a continuación. Además, cabe destacar que los filtros de Kalman actúan tras la acción del estimador dentro del sistema y justo antes del proceso de representación de dicha estimación, para que sea finalmente, la orientación proporcionada por el filtro de Kalman, la que se represente en la imagen estimada.

Por último, se ha realizado un proceso de validación que afecta a la inicialización y a la configuración de parámetros para cada modelo de tracking basado en el filtro de Kalman. Esta fase de validación se ha basado en un triple proceso de ajuste, uno por cada ángulo a estimar en $\hat{\theta}$, ya que cada uno es procesado por un filtro independiente de tracking. Primero tomamos el HF+PLEV $\mathcal{F}$ entrenado, y se elige un conjunto de imágenes de entrenamiento para realizar el proceso de ajuste de los parámetros de tracking. Para el KF, el paso de validación consiste en ajustar la inicialización para las siguientes matrices: a) covarianza del proceso de ruido (Q), b) covarianza de la medida de ruido (R), c) covarianza de la estimación del error posterior (P) y d) medida (H) [33]. Inicializamos el primer estado predicho del KF de acuerdo a $\hat{\theta}$.

2.3.2. Filtro de Partículas: Modelo e Implementación

El filtro de partículas o algoritmo de CONDENSACIÓN (Propagación Condicional de Densidad) se basa en técnicas de muestreo, para su aplicación de forma iterativa en imágenes contenidas en una secuencia. El proceso consiste en una iteración de muestreo factorizando cada paso de tiempo que sucede al siguiente [19]. Como resultado tras la iteración, se obtiene una muestra temporal caracterizada por un peso, la cual se representa como $\{s_t^{(n)}, n = 1, .., N\}$, caracterizando a cada muestra con un peso $\pi_t^{(n)}$, y cuya densidad condicional de estado del objeto, en un instante concreto, es representada como $p(x_t|Z_t)$ en el instante t. Esta ecuación representa la información acerca del estado de la cabeza en el instante t, que se puede deducir de todo el flujo de datos obtenido de la trayectoria de la misma antes del instante de tiempo actual. Por lo que se puede denominar a x_t como el estado del objeto en el instante t, y a Z_t como el histórico de las observaciones realizadas sobre la imagen hasta un instante t. Además, cabe destacar que N es el conjunto de muestras escogidas. Atendiendo a la obtención de cada densidad condicional del objeto, en este caso de la de los 3 ángulos que conforman la orientación de la pose de la cabeza, se sigue el siguiente proceso; para comenzar se dispone de una densidad inicial con lo que se dispone de $p(x_t|Z_{t-1})$.

El funcionamiento del proceso iterativo del algoritmo es el siguiente. Para comenzar, se muestrean y reemplazan entre ellos, aquellos elementos del conjunto $s_{t-1}^{(n)}$ con una probabilidad $\pi_{t-1}^{(n)}$. La selección de la muestra $s_t^{'(n)}$ comienza tras la elección de un número aleatorio $r \in [0, 1]$. Tras ello, se produce una subdivisión binaria, donde se escoge el menor j para el cual $c_{t-1}^{(j)} \geq r$. Por último, se fija el dato $s_t^{'(n)} = s_{t-1}^{(j)}$. De este paso inicial se deduce, que aquellas muestras que dispongan de una mayor probabilidad asignada pueden ser elegidas en varias ocasiones, y aquellas que dispongan de menores probabilidades puede que nunca sean escogidas dentro del proceso de muestreo.

A continuación, aquellas muestras seleccionadas son sometidas al proceso de predicción. En el caso de este trabajo, se ha hecho uso de una ecuación estocástica lineal para el modelado de la obtención de las muestras del conjunto de datos en el nuevo instante de tiempo:

$$s_t^{(n)} = As_t^{'(n)} + Bw_t^{(n)}, \tag{2.16}$$

donde w_t es el vector de variables normales independientes, s_t es el vector de estado y A y B las matrices de área que representan las componentes determinística y estocástica, del modelo dinámico, respectivamente.

Esta última ecuación, modela una corriente de muestras donde aquellas muestras que son idénticas tienden a dirigirse en el mismo flujo de desplazamiento. Tras ello, se realiza el proceso de difusión, el cual es un proceso totalmente aleatorio donde aquellas muestras idénticas dentro del mismo flujo de desplazamiento se separan. Tras finalizar este paso, se dispone del conjunto de datos $\{s_t^{(n)}\}$ que hace referencia al siguiente instante de tiempo, faltando únicamente la obtención del vector de pesos.

Por último, se realiza la etapa de observación donde se aplica la técnica de muestreo factorizado para la asignación de pesos a las muestras del conjunto de datos, a partir de la densidad $p(z_t|x_t)$, realizando la siguiente asignación $\pi_t^{(n)} = p(z_t|x_t = s_t^{(n)})$, para obtener el conjunto de datos de las densidades de estado en el instante t, $\{(s_t^{(n)}, \pi_t^{(n)}, c_t^n)\}$. Se puede ver incluido en el conjunto de datos resultante de la iteración en el instante t, la presencia de la probabilidad acumulativa, cuyo valor es $c_t^0 = 0$ y $c_t^n = c_t^{(n-1)} + \pi_t^{(n)} (n = 1, ..., N)$. Un resumen de las 3 etapas descritas durante esta sección se puede observar en la Figura 2.4 [19].

Una vez se han obtenido N muestras, se realiza la estimación de pose, en el instante t, donde para cada ángulo se aplica la siguiente ecuación de estimación:

$$\varepsilon[f(x_t)] = \sum_{n=1}^{N} \pi_t^{(n)} f(s_t^{(n)}), \tag{2.17}$$

donde se observa como se computan las respectivas muestras de estado con respecto a su peso correspondiente. Posteriormente se obtiene el valor medio del valor de cada ángulo en un determinado instante de tiempo mediante la asignación del estado del objeto a cada instante de tiempo, a través de $f(x_t) = x_t$.

En cuanto a la implementación, al igual que para el filtro de Kalman, se ha incluido en el sistema este filtro de partículas justo tras la realización del proceso de estimación, cuyos valores son las muestras de entrada a dicho filtro.

Con el PF, se ha realizado la validación para ajustar los siguientes parámetros: a) el número de muestras generadas por el filtro (N), b) el número de iteraciones, y c) los parámetros característicos del modelo dinámico ($A, \bar{x}, B$). Esto se ha realizado llevando a cabo un proceso de ajuste, donde se han testeado diferentes combinaciones de valores,

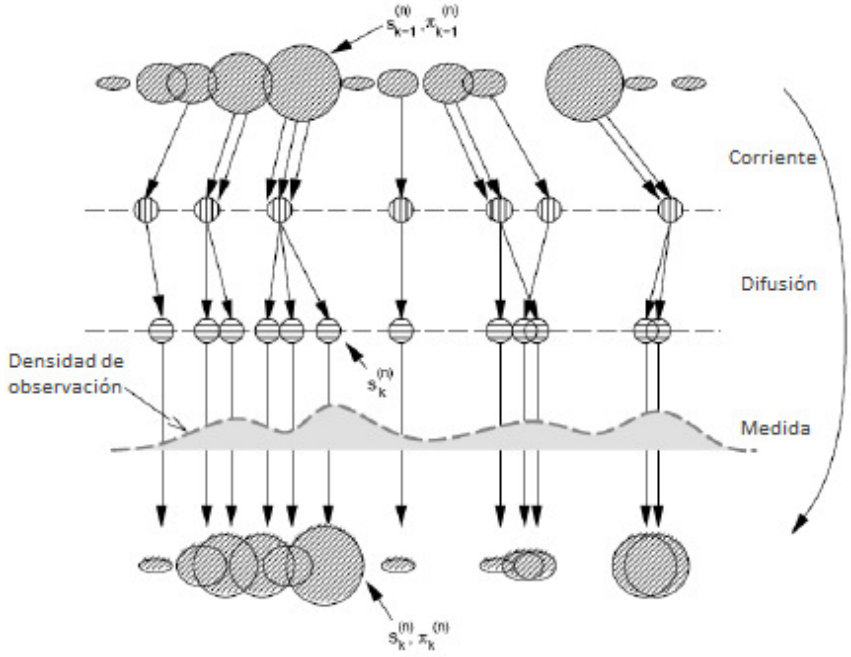


Figura 2.4: Proceso de estimación de un conjunto de muestras en un instante de tiempo.

habiendo sido escogidos aquellos que proporcionan unos mejores resultados, en cuanto a la reducción del error en la estimación de la orientación de la pose.

2.4. Resultados

2.4.1. Descripción del experimento

Conjunto de datos empleado. Reportamos el rendimiento del modelo haciendo uso únicamente de imágenes RGB procedentes de la base de datos *Biwi Kinect Head Pose Database* (Biwi) [9]. Contiene alrededor de 15K imágenes de 20 personas. El rango de pose de cabeza cubre alrededor de ±75 grados de dirección y ±60 grados de elevación. El valor real es proporcionado en forma de localización 3D de la cabeza y su rotación. Siguiendo [9], dividimos la base de datos en 2 conjuntos:un conjunto de test y de entrenamiento de respectivamente 2 (sujetos 1 y 12) y 18 sujetos. Se hace uso del sujeto de entrenamiento 24 para la validación.

Detalles de implementación. Se ha desarrollado el sistema en C++ de forma íntegra, portando todas las funciones de Matlab del HF+PLEV original [26].

Durante el entrenamiento, los ejemplos positivos son recortados y rescalados al mismo tamaño, elegido para que la dimensión mas larga de la caja contenedora sea de 100 píxeles. 20 parches positivos y 20 negativos, con un tamaño de 32×32 píxeles, son extraídos de forma aleatoria de cada imagen de entrenamiento. Nuestros bosques disponen de 15 árboles con una profundidad máxima de 20. En cada nodo, 20,000 tests binarios son considerados durante el aprendizaje. Para el PLEV consideramos un tamaño de vecindad

Tabla 2.1: Parámetros óptimos de configuración calculados para los filtros de tracking.

Parámetros óptimos de los filtros de tracking		
Filtro	Parámetro	Valor
Partículas	$N_{muestras}$	1300
	$N_{iteraciones}$	500
	$\bar{x}_{direccin}$	0.1
	$\bar{x}_{elevacin}$	0.15
	$\bar{x}_{alabeo}$	0.15
	$A_{direccin}$	0.5
	$A_{elevacin}$	7
	A_{alabeo}	0.5
	$B_{direccin}$	0.079
	$B_{elevacin}$	0.3
	B_{alabeo}	0.079
Kalman	H	I
	Q	$I10^{-4}$
	$R_{direccin}$	$I10^{-1}$
	$R_{elevacin}$	$I10^{4}$
	R_{alabeo}	$I10^{3}$
	$P_{direccin}$	$I10^{1}$
	$P_{direccin}$	$I10^{8}$
	P_{alabeo}	$I10^{5}$

de 11×11 píxeles. Además, se hace uso de los 32 canales de característica empleados en [13].

Para el detector de color de piel, se ha aplicado una conversión de imagen al espacio de color YCrCb. Las restricciones del filtro de color permiten reconocer como píxeles de piel aquellos que cumplen: Y$\in$ $[15, 235]$,Cb$\in$ $[60, 122]$,Cr$\in$ $[135, 170]$. Se realiza un filtrado de todas las regiones del candidato, descartando aquellas regiones cuyo área en píxeles se encuentre fuera del intervalo $[4000, 40000]$, o cuya relación de aspecto ($\frac{altura}{anchura}$) no pertenezca al intervalo [0,5, 3]. Continuando con el proceso de validación descrito para los parámetros de filtrado de tracking, los valores elegidos se pueden observar en la Tabla 2.1.

2.4.2. Resultados de la detección de cabeza y estimación de pose

Los resultados son reportados en la Tabla 2.2, ofreciendo una comparación entre el estado del arte usando imágenes RGB y de profundidad.

Primero de todo, es importante mencionar que la estimación para la dirección, la elevación y el alabeo, reduce el error con respecto a [26], incluso para una tasa inferior de

Tabla 2.2: Resultados empleando la Biwi Kinect Head Pose Database.

Imágenes	Modelo	Posición Error (mm)	Dirección Error Total (°)	Dirección (°)	Elevación (°)	Alabeo (°)	Perdidos (%)
Profundidad	[27]	$8{,}1 \pm 5{,}3$	$9{,}8 \pm 8$	$\mathbf{3{,}8 \pm 3{,}7}$	$6{,}7 \pm 6{,}6$	$4{,}3 \pm 4{,}9$	**1**
	[28]	$10{,}8 \pm 6{,}1$	$12{,}2 \pm 9$	$5{,}5° \pm 5{,}8°$	$7{,}8° \pm 7{,}9°$	$5° \pm 4{,}4°$	3
	[26]	$\mathbf{7{,}2 \pm 12{,}1}$	$7{,}3 \pm 5{,}9$	$4{,}1° \pm 6{,}9°$	$3{,}9° \pm 4°$	$3{,}2° \pm 3°$	5
	[9]	$12{,}2 \pm 22{,}8$	$\mathbf{5{,}9 \pm 8{,}1}$	$3{,}8 \pm 6{,}5°$	$\mathbf{3{,}5 \pm 5{,}8°}$	$5{,}4 \pm 6{,}0°$	6.6
	[11]	$14{,}7 \pm 22{,}5$	–	$9{,}2 \pm 13{,}7°$	$8{,}5 \pm 10{,}1°$	$8 \pm 8{,}3°$	**1**
Imágenes	Modelo	Posición Error (píxeles)	Dirección Error Total (°)	Dirección (°)	Elevación (°)	Alabeo (°)	Perdidos (%)
RGB	[26]	$3{,}2 \pm 1{,}4$	$9{,}8 \pm 6{,}8$	$5{,}8 \pm 5{,}9$	$5{,}8 \pm 4{,}8$	$3{,}5 \pm 3{,}4$	2.4
	Nuestro + KF	$\mathbf{3 \pm 1{,}4}$	$9{,}4 \pm 7{,}5$	$5{,}8 \pm 6{,}4$	$5{,}1 \pm 4{,}6$	$3{,}1 \pm 3{,}3$	1.4
	Nuestro + PF	$\mathbf{3 \pm 1{,}4}$	$9{,}1 \pm 6{,}9$	$5{,}3 \pm 6{,}3$	$5{,}2 \pm 4{,}4$	$\mathbf{2{,}8 \pm 2{,}9}$	1.4

fotogramas perdidos (1,4 %). Además, nuestro modelo, haciendo uso del mismo bosque entrenado que [26], y considerando que el tracking es realizado únicamente para las estimaciones de pose (no para la localización de la cabeza), alcanza un rendimiento en la detección de rostro ligeramente superior que en [26]. Los resultados muestran que el KF es capaz de reducir en gran medida los errores en elevación y alabeo. El PF nos permite reducir todos los errores de pose, y especialmente aquellos asociados con el alabeo. Experimentalmente, se ha observado que el PF consigue una precisión ligeramente superior a la lograda por el KF. En conclusión, nuestra aproximación supera el estado del arte reportado en [26], tanto en lo relativo a la detección de rostro como a la estimación de pose de cabeza, cuando además solo se han empleado imágenes RGB.

Notar que nuestro modelo emplea simples datos RGB, superando incluso los resultados de métodos que necesitan imágenes de profundidad, como [26, 28, 27]. Es relevante apuntar que nuestro error para las estimaciones de alabeo es el mas bajo en ser reportado haciendo uso de este conjunto de datos. Atendiendo a los errores en dirección y elevación, el incremento de nuestro error es de 1,5° y 1,7°, respectivamente, comparado con los métodos ganadores que hacen uso de la profundidad. Para el error en la dirección de la nariz, nuestro método se encuentra lejos de [11], pero se debe notar que su ratio de fotogramas perdidos es mayor. Para un ratio comparable de fotogramas perdidos, *p.ej.* [27], nuestro método ofrece una estimación de la dirección de la nariz ligeramente mejor.

Como conclusión, se puede afirmar que nuestro modelo es capaz de mejorar los resultados del estado del arte para el problema de la estimación de pose de cabeza en imágenes RGB, lo que es una importante contribución. Los resultados cualitativos se proporcionan en la Figura 2.6.

Nuestro modelo ha sido diseñado para realizar una rápida estimación para el HCI, mientras que la precisión no se ve degradada. En el siguiente experimento, se evalúa el rendimiento de nuestra aproximación como función del parámetro de salto de píxeles. Recordar que este parámetro controla el muestreo espacial de los parches a ser extraídos en una región de entrada de un candidato identificada por el paso del detector de piel. Este parámetro se puede ajustar para definir el balance deseado entre precisión y velocidad de

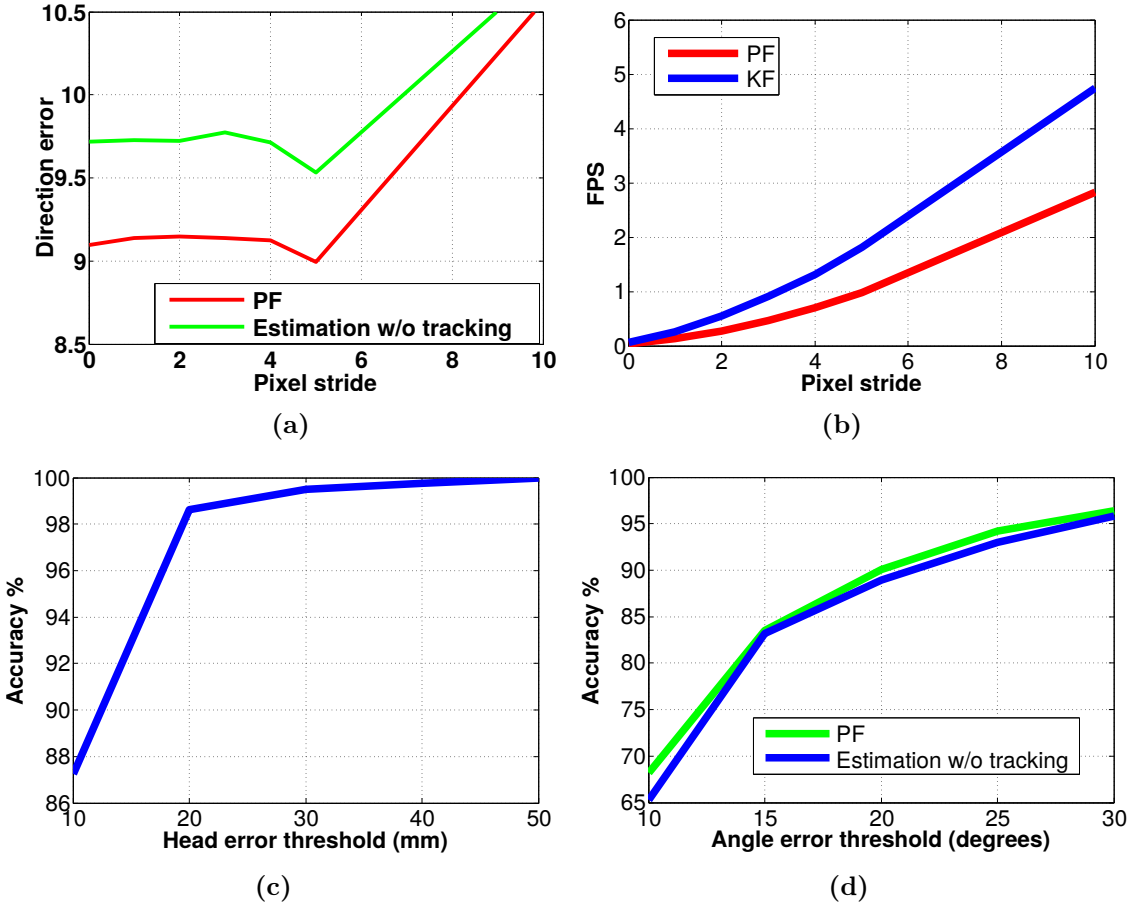


Figura 2.5: (a) y (b) resultados cuantitativos como función del parámetro de salto de píxeles. (a) Errores para la dirección de la nariz. (b) Fotogramas Por Segundo (FPS). Precisión de fotograma de nuestro método para diferentes umbrales de éxito. (c) La posición de la cabeza en mm y (d) La pose de la cabeza en grados.

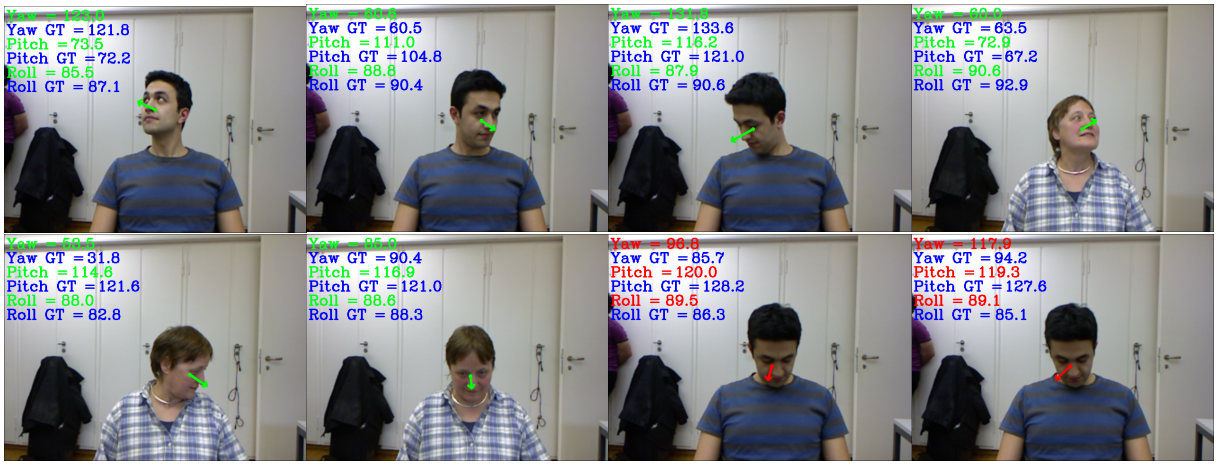


Figura 2.6: Resultados cualitativos. Valor real en azul, estimaciones buenas en verde y estimaciones erróneas en rojo.

ejecución del proceso. La Figura 2.5(a) muestra como el error en la dirección varía con el salto de píxeles. Podemos observar como nuestro modelo con PF reporta sistemáticamente mejores resultados que nuestra implementación sin tracking. Esto revela que la solución de tracking implementada mejora realmente el rendimiento del sistema. La Figura 2.5(b) muestra los fotogramas por segundo de nuestra implementación para diferentes valores de salto de píxeles. Es relevante destacar que nuestra aproximación es capaz de procesar hasta 3 fotogramas por segundo, haciendo uso del PF, reportando un error de dirección inferior a 11°.

Finalmente, se reporta la precisión como función del umbral de éxito para el error de la localización de la cabeza y el error en la estimación de pose en las Figuras 2.5(c) y 2.5(d), respectivamente. Podemos observar (ver Figura 2.5(c)) que nuestro modelo proporciona un buen rendimiento para todos los umbrales de éxito en la regresión de la posición de la cabeza. Alcanza un 87,24 % para el umbral mas restrictivo, el de 10 mm. Con respecto al umbral del error de ángulo empleado para evaluar la estimación de pose, la Figura 2.5(d) muestra que el tracking con PF mejora la precisión del modelo. Es importante mencionar que para un umbral restrictivo de 10°, nuestra implementación con tracking reporta un 68,2 % de precisión, comparado con el 65 % de nuestra aproximación sin la etapa de tracking.

Capítulo 3

Sistema de detección de carriles de circulación mediante estimación de pose de vehículos

3.1. Modelo teórico de la estimación de pose en vehículos

En este capítulo se van a analizar los experimentos realizados y los resultados obtenidos en lo relativo a la estimación de pose en vehículos, y en concreto a su aplicación a la estimación de carriles.

En este caso, el HF es empleado para la detección y estimación de pose, al igual que en el caso de estimación de pose en cabezas.

En cuanto a la etapa de entrenamiento, el proceso ha sido realizado de forma análoga al caso anterior, con la diferencia de que los árboles del HF han sido entrenados con imágenes de vehículos. En estas imágenes, el vehículo dispone de un amplio tamaño de píxeles, y la resolución de dichas imágenes es mayor que las habituales imágenes de tráfico, por lo que será posteriormente, antes de lanzar los parches de imágenes por los RF durante la etapa de test, cuando se realicen ajustes de resolución de cada fotograma de entrada al sistema. Además, cabe destacar que estas imágenes disponen de un vector de localización $\theta^1 = (\theta_x, \theta_y)$, por lo que se nota la eliminación de la componente z de dicho vector, ya que la información proporcionada por esta coordenada no se considera relevante para tareas de estimación. En cuanto al vector de orientación, $\theta^2 = (\theta_{acimut}, \theta_{zenit})$, se puede observar como las imágenes han sido anotadas con información de orientación relativa a los ángulos de zenit y acimut, los cuales van a describir la trayectoria que siguen dichos vehículos en la imagen. Además, de forma adicional, se ha anotado la anchura y altura medias de las cajas contenedoras de los vehículos en las imágenes de entrenamiento, para poder ser empleada durante la etapa de test en la diferenciación clara de los diferentes vehículos estimados.

En lo relativo al proceso de test, la diferencia con respecto a la sección anterior es la presencia del proceso de *backprojection* o retroproyección [26], tras finalizar la estimación en los RF. Esta estrategia se emplea para la estimación de cajas delimitadoras, que contienen a los vehículos estimados, de la siguiente forma. Para cada hipótesis de objeto $\hat{h}$, se realiza la retroproyección sobre la imagen, de la caja contenedora mas grande de las imágenes de entrenamiento. En el interior de esta caja, los parches son extraídos de forma densa para posteriormente ser lanzados a través del HF. Para computar la máscara de retroproyección, cada vez que un parche en la posición x, por ejemplo $P(x)$, vota por $\hat{h}$, se computa su peso de contribución $\omega(P(x), \hat{h})$ como se indica en la siguiente ecuación:

$$\omega(P(x), \hat{h}) = \frac{1}{T} \sum_{t=1}^{T} \left(\frac{p(c = 1|L_t(P(x)))}{|L_t(P(x), c = 1)|} \sum_{L_t(P(x),c=1)} K(\hat{h}, d, \theta, x) \right), \tag{3.1}$$

donde $L_t(P(x))$ es la hoja alcanzada por el parche $P(x)$, y, $p(c = 1|L_t(P(x)))$ y $|L_t(P(x), c = 1)|$ codifican la probabilidad de objeto y el número de parches de objeto en la hoja $L_t(P(x))$, respectivamente. Además, se debe notar la presencia de los vectores de localización d y orientación θ, así como de la hipótesis a tratar $\hat{h}$ y la coordenada seleccionada para votar por dicha hipótesis. También se hace uso de una contribución de pesos modificada [26], empleando la localización en 2 dimensiones d y la información de la pose θ en la hoja $L_t(P(y))$. Se penaliza a aquellos parches que votan por diferentes poses, además de por diferentes localizaciones. Para ello, se define la componente $K(\hat{h}, d, \theta, x)$ como se muestra a continuación.

$$K(\hat{h}, d, \theta, x) = exp\left(-\sqrt{\frac{1}{\lambda}||d - \frac{u(x-\hat{h}_d)}{s_i}||^2 + \left(\frac{min\{||\hat{\theta}-\theta||, 360^o - (||\hat{\theta}-\theta||)\}}{180^o} \right)^2} \right), \tag{3.2}$$

donde λ es el parámetro normalizador igual a $(u - R)^2$, u es el tamaño normalizado de las cajas contenedoras en la etapa de entrenamiento, y R es el tamaño del parche. El primer término de la ecuación hace referencia al error en la estimación de la localización y el segundo es aplicado a la pose. Para obtener la caja contenedora, la máscara es umbralizada para estimar la caja mas ajustada que ocupa la máscara binaria. Dicha umbral en [26] es definido como $\frac{1}{2} max_x(\omega(P(x), \hat{h}))$.

Por último cabe destacar la descripción del modelo empleado para la detección de carriles, a partir de la información proporcionada por la estimación de pose de los vehículos en imágenes de tráfico. La agrupación jerárquica ha sido la técnica empleada para realizar la agrupación de vehículos con similitudes, en función de la distancia euclídea medida en píxeles entre los vehículos, incluyendo también la información del ángulo del zenit. Esta técnica ha sido escogida frente al método K-means de agrupación, dado que, aunque los resultados proporcionados por esta técnica no se consideran incorrectos, existe una limitación dado que es necesario especificar a las funciones de dicho método, el número

de carriles que se deben identificar en la imagen, lo cual no es factible en nuestro trabajo, dado que dependiendo de la calzada, el sistema puede analizar mas de 2 carriles en una misma imagen, en caso de que existan cruces, por ejemplo.

La agrupación jerárquica se basa en la creación de dendrogramas, un diagrama de organización de datos en forma de árbol, viéndose el detalle de los datos aumentado, a medida que aumenta la subdivisión de dicho árbol. En este caso, las agrupaciones son ordenadas en estos dendrogramas.

Para hacer uso de la agrupación jerárquica, se ha comenzado por calcular la distancia euclídea entre cada centro estimado presente en la imagen con el resto de centros. En la Figura 3.1, se puede observar primero la imagen a partir de la cual se realiza la detección de carriles. Cabe destacar que solo serán tenidos en cuenta aquellos centros estimados que se encuentren dentro de la región de interés definida para dicho fin. En la Figura 3.1, también se puede observar como la distancia que aparentemente es menor entre los vehículos del fondo de la imagen, se incrementa en nuestro algoritmo, dado que se tiene en cuenta el valor del ángulo del zenit, el cual es muy diferente entre los vehículos del fondo de la imagen, lo cual produce que ambos carriles sean identificados como diferentes.

Posteriormente, se realiza una agrupación de pares de vehículos basándonos en agrupar aquellos vehículos que presentan mayor proximidad entre sí, basándonos en la distancia euclídea entre los mismos. Estas agrupaciones de pares de vehículos son almacenados en el vector de información de distancia Y. De esta forma, se van construyendo pares de agrupaciones binarias, y se va formando un árbol jerárquico o dendrograma, donde se van agrupando estos grupos binarios a medida que crece la profundidad de dicho árbol. En la Figura 3.2 se pueden observar las agrupaciones realizadas por el sistema, para distinguir aquellos vehículos que por proximidad u orientación, tienen mayores probabilidades de encontrarse circulando en el mismo carril de la calzada.

Por último, se ha indicado la distancia euclídea umbral de 350 para determinar que vehículos deben mantenerse dentro de la misma agrupación, o lo que es lo mismo, se han determinado aquellas ramas del dendrograma que dejarán de crecer y que presentarán las agrupaciones de vehículos de forma independiente por cada rama. Cabe destacar que este valor umbral de 350 incluye la suma entre la distancia euclídea en píxeles entre los vehículos y la diferencia entre la información de orientación que presenta cada vehículo. Tras aplicar, esta distancia umbral definida para el sistema, se verán diferenciados los diferentes grupos de vehículos que pertenecen a carriles de circulación diferentes.

Cabe destacar que en el sistema se ha incluido una estimación de carriles aglomerativa, lo cual suma las diferentes estimaciones de carriles realizadas, a lo largo del tiempo, sobre las imágenes procedentes de la misma cámara de tráfico. Esto se realiza para cubrir todo el terreno de calzada posible con la representación de cada carril, ya que en muy pocas imágenes, existen vehículos que cubran todas las zonas de la calzada en todos los carriles de circulación.

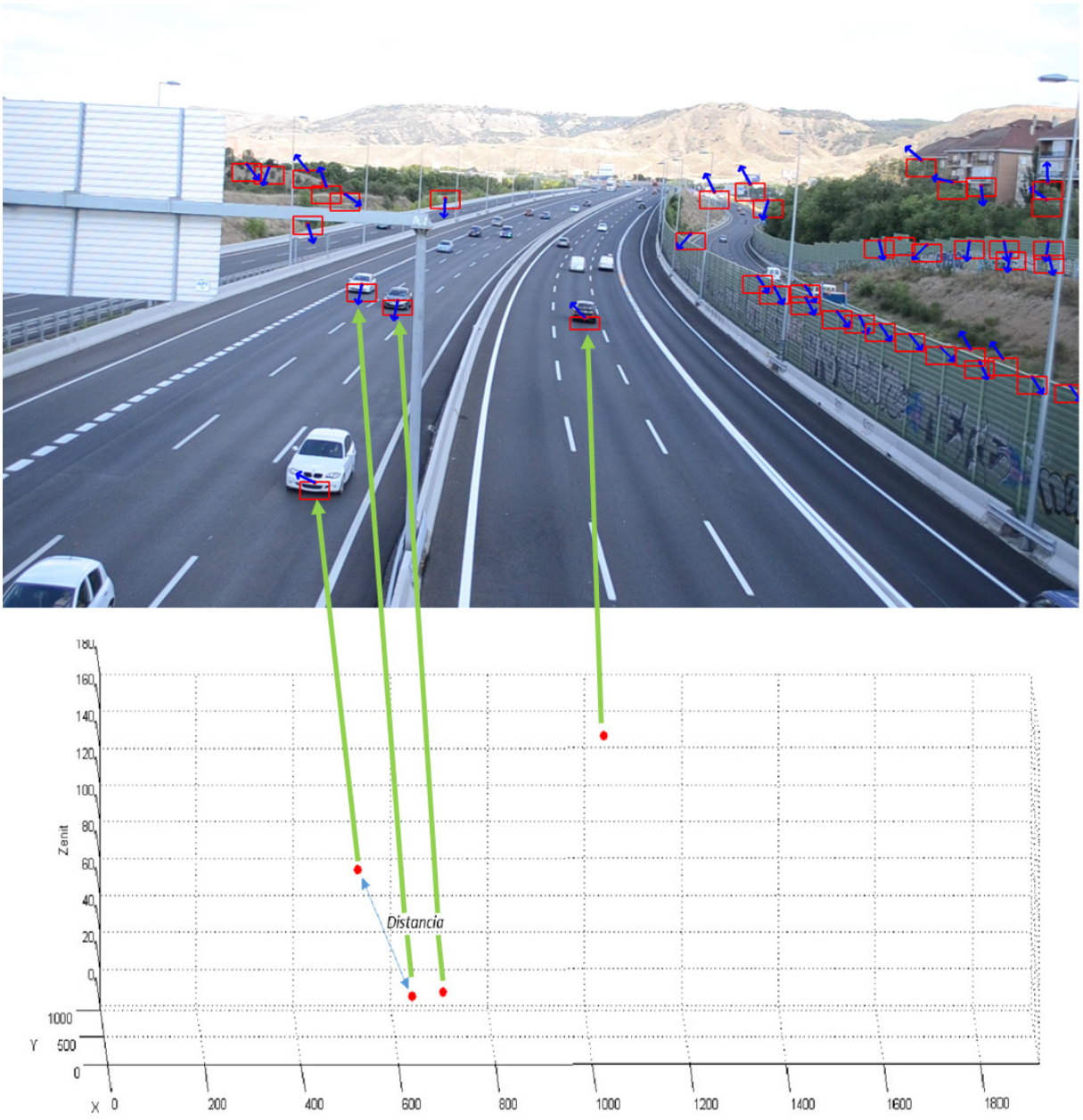


Figura 3.1: Relación entre la imagen de entrada al detector de carriles, incluyendo las estimaciones realizadas sobre la misma, y la imagen en 3 dimensiones del cálculo realizado para definir la distancia entre centros.

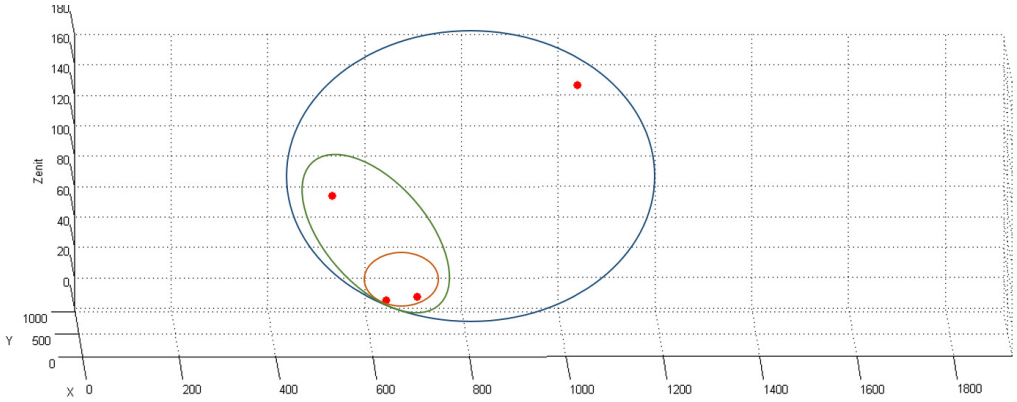


Figura 3.2: Agrupaciones formadas por el algoritmo del sistema, teniendo en cuenta la distancia en píxeles así como la diferencia entre ángulos de orientación.

Figura 3.3: Separación por grupos de vehículos pertenecientes a diferentes carriles de circulación, realizada por el sistema.

3.2. Resultados

3.2.1. Descripción del experimento y las bases de datos

Se han realizado experimentos con dos conjuntos de imágenes de tráfico. El primer conjunto ha sido tomado de la base de datos TRANCOS [17], y son 1244 imágenes procedentes de las cámaras de tráfico de la DGT ubicadas en carreteras de la Comunidad de Madrid, con una resolución de 640x480 píxeles, donde se dispone del número de vehículos presentes en cada fotograma (en total 46796 vehículos anotados en toda la base de datos), así como de los centros de los vehículos anotados [17] para poder analizar el error cometido por nuestro estimador en cuanto a localización. Cabe destacar que las imágenes contenidas en esta base de datos están caracterizadas por una alta congestión de vehículos, en al menos, uno de los carriles de circulación. Además, de forma adicional a los centros anotados en cada fotograma, se dispone de una Región de Interés (ROI) anotada, también por cada fotograma, ya que en esta base de datos se recogen imágenes de distintas cámaras de la DGT. La función de esta máscara es delimitar las zonas de la imagen que forman parte de la calzada, tratando de seleccionar aquellas partes que puedan ser empleadas para tareas de conteo y estimación de vehículos, ya que en ocasiones las zonas mas lejanas de la calzada que pueden ser observadas desde la cámara no son seleccionadas, dado que los vehículos apenas pueden ser observados en tal zona. Para la etapa de entrenamiento con esta base de datos fueron empleadas 403 imágenes, para validación 420 imágenes y para la etapa de testeo 421 [17]. A continuación, se pueden observar algunas de las imágenes de la base de datos TRANCOS en las Figuras 3.5(a), 3.5(b), 3.5(c).

Por otro lado, se ha construido una base de datos con 213 imágenes de tráfico de una mayor resolución, 1689x942 píxeles, para testear las mejoras que introduce una imagen de mayor calidad a la hora de reducir el error de vehículos estimados y su pose. Se debe notar una alta presencia de estimaciones que no se corresponden con zonas de la calzada, dado que el entorno que rodea la carretera produce un efecto ruidoso en el estimador. Esto en cambio, no produce ninguna clase de problema, ya que se hace uso de unas máscaras preconstruidas, una vez se conoce la situación y forma de el tramo total de vía. Con esto, se consiguen filtrar las zonas de la imagen que no se corresponden con la calzada para que éstas no sean tenidas en cuenta a la hora de realizar la detección de carriles. En la Figura 3.4 se muestran la máscara empleada para aplicar la estimación únicamente sobre la región de interés que aplica a este Trabajo de Investigación.

Algunas imágenes procedentes de esta base de datos, se pueden encontrar en las Figuras 3.5(d), 3.5(e), 3.5(f). Es importante remarcar el hecho de que, el sistema propuesto funciona correctamente sin necesidad de testear vídeos, formados por fotogramas consecutivos, lo cual podría dar una información adicional que facilitaría las tareas de estimación, pero de la cual no se dispone habitualmente, dado que normalmente, las cámaras de la DGT se actualizan con un nuevo fotograma cada 5 minutos. Por ello, este estimador está pensado para funcionar correctamente con fotogramas independientes, donde la única

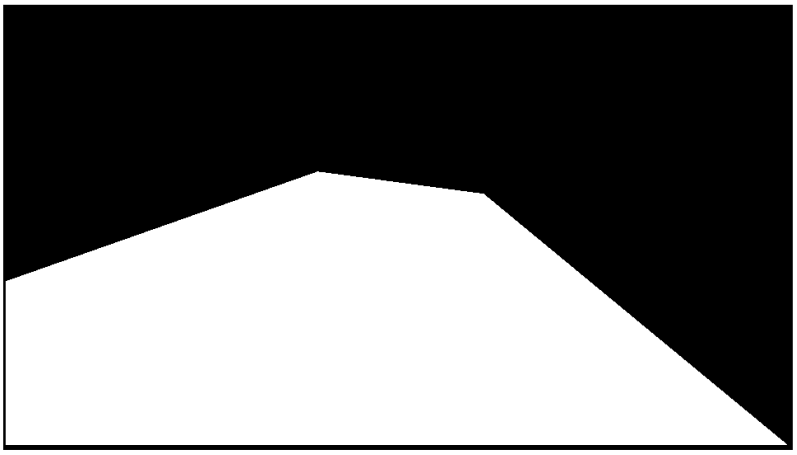


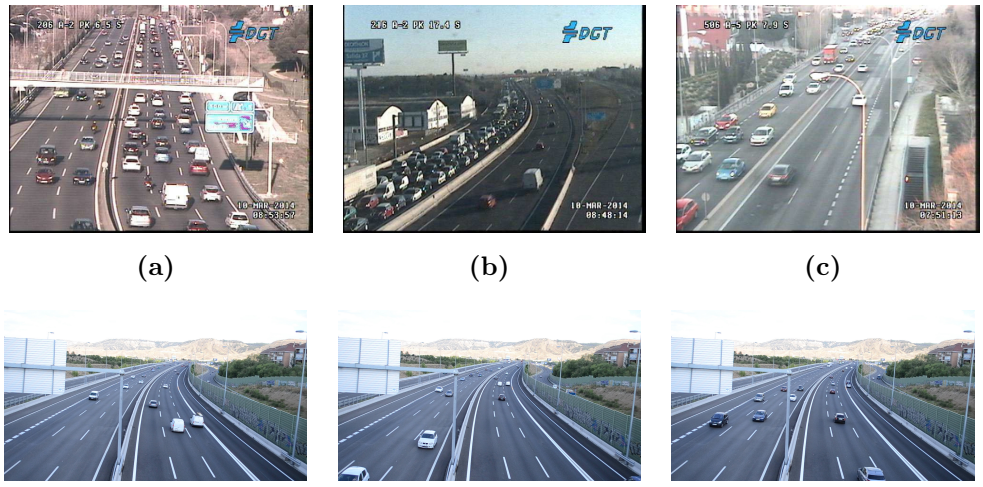
Figura 3.4: Máscara generada para el filtrado de detecciones en la base de datos de imágenes de tráfico generada.

información a procesar se encuentra presente en dicha imagen. Como se ha comentado anteriormente, la información temporal que proporciona un vídeo no siempre está disponible en instalaciones complejas de videovigilancia. Además, la estimación de un solo fotograma puede proporcionar información similar a la que presentaría la opción de estimación en vídeo, pero teniendo que procesar muchos menos fotogramas en el tiempo, por lo que la estimación final se obtendría de forma mas rápida.

Los árboles del modelo de detección y estimación de pose han sido entrenados en imágenes con vehículos, al igual que en el caso de estimación de pose de cabezas, para cajas contenedoras de 100 píxeles, siendo el número de árboles (15) y su profundidad la misma que el bosque aleatorio empleado en el caso anterior. Dicho entrenamiento se ha realizado empleando la base de datos de vehículos de Glasner *et al* [16], donde el tamaño de los vehículos es mayor que el que se presenta en las imágenes de las cámaras de tráfico en carretera. En la Figura 3.6 se pueden observar 2 imágenes procedentes de dicha base de datos, donde se ve como el tamaño de los vehículos es bastante superior al de las imágenes de tráfico.

En este caso, dado que se hace uso de imágenes de tráfico en diferentes situaciones, tanto en términos de volumen de tráfico en cada fotograma, como de resolución de imágenes dependiendo de la base de datos que se ha estimado, hay dos parámetros con una importancia notable en el éxito de los resultados obtenidos que son, el valor de escalado empleado a la hora de estimar la imagen y el valor de Non-Maximum Supression (NMS), que controla la posibilidad de estimar varios centros que se encuentren próximos entre si (a mayor valor de NMS, mayor debe ser la distancia entre vehículos para que el estimador permita reconocer varios centros como válidos). Su influencia se comentará en los próximos resultados.

En lo relativo a la estimación de carriles, se hace uso de la información obtenida tras el



(a) **(b)** **(c)**

(d) **(e)** **(f)**

Figura 3.5: Parte superior de la figura: Imágenes procedentes de la base de datos TRANCOS. Parte inferior de la figura: Imágenes procedentes de la base de datos propia generada.

(a) **(b)**

Figura 3.6: Imágenes estimadas procedentes de la base de datos [16].

proceso de estimación, en lo relativo a localización del vehículo, mediante las coordenadas x e y, así como de orientación, haciendo uso del ángulo del zenit. Cabe destacar que el número de carriles de circulación que nuestro sistema es capaz de estimar no dispone de ningún límite, por lo que el sistema es capaz de estimar tanto imágenes de 2 carriles de circulación en autovía, como la identificación de cruces e incorporaciones y salidas en carretera.

3.2.2. Resultados en la base de datos TRANCOS

En cuanto a los resultados del estimador para la estimación de vehículos y la posterior detección de carriles, se pueden clasificar en cuantitativos y cualitativos. Comenzando por los cuantitativos, se ha realizado un estudio de análisis del error en la estimación de centros de vehículos, para la base de datos de baja resolución de imágenes procedentes de la DGT [17], haciendo uso de la métrica de evaluación GAME (Grid Average Mean absolute Error) [17],

Para conocer el funcionamiento de la métrica GAME [17], se debe aprender primero como emplear la métrica MAE (Mean Absolute Error).

$$\mathcal{MAE} = \frac{1}{N} * \sum_{n=1}^{N} |e_n - gt_n|, \tag{3.3}$$

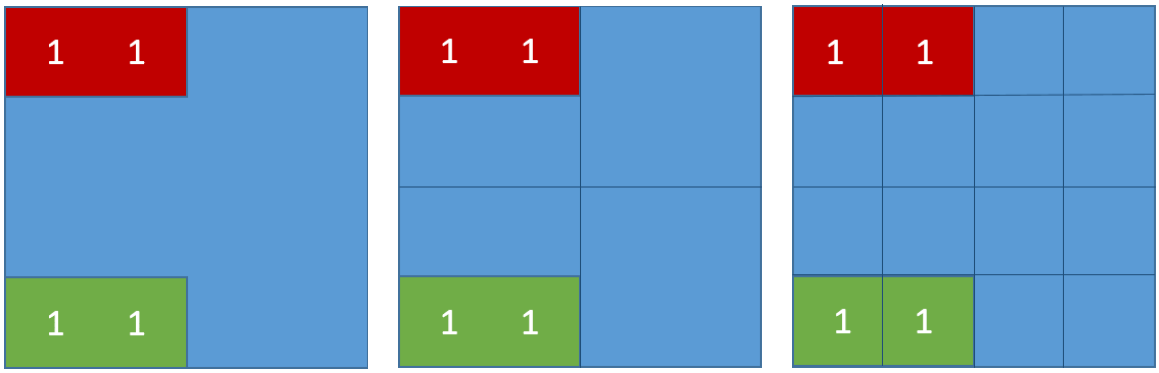
donde, e_n se corresponde con los objetos contados en la imagen n, gt_n hace referencia al valor real de los objetos presentes en la imagen, y N se corresponde con el total de imágenes tenidas en cuenta para el cálculo. El problema de esta métrica, es que no tiene en cuenta la posición de los objetos contados. Por ello, se hace uso de la nueva métrica, que tiene en cuenta tanto el conteo de objetos como la localización estimada de cada uno de ellos. La imagen es subdividida en 4^L regiones no superpuestas y se realiza el cálculo del MAE en cada una de estas regiones. El GAME tiene la siguiente forma matemática,

$$\mathcal{GAME}(\mathcal{L}) = \frac{1}{N} * \sum_{n=1}^{N} (\sum_{l=1}^{4^L} |e_n^l - gt_n^l|), \tag{3.4}$$

donde, e_n^L es la cuenta de objetos en la región l de la imagen n, y gt_n^l es el valor real para la misma región de la misma imagen. Cuanto mayor es el valor de L, mas restrictiva es la métrica GAME, destacando que la métrica MAE es una particularización del GAME para un valor de $L = 0$.

En la Figura 3.7, se puede observar de forma gráfica el funcionamiento de este algoritmo. En la Figura 3.7(a), se observa como el cálculo del MAE, no muestra ningún error en cuanto a los 2 vehículos detectados (estimaciones en verde) con respecto a su ubicación real (estimaciones en rojo). En cambio, se observa como en las Figuras 3.7(b) y 3.7(c), el GAME penaliza estos errores en la detección, ya que el algoritmo tiene en cuenta la localización de las detecciones de forma adicional al número de vehículos estimados.

En la Tabla 3.1 mostramos los resultados obtenidos por nuestro sistema, y una comparación con el estado del arte en esta base de datos. Nuestro estimador queda en último



(a) **(b)** **(c)**

Figura 3.7: a) Cálculo del error a través del MAE, MAE = 0. b) Cálculo del error a través de GAME(1), GAME(1) = 4. c) Cálculo del error a través de GAME(2), GAME(2) = 4.

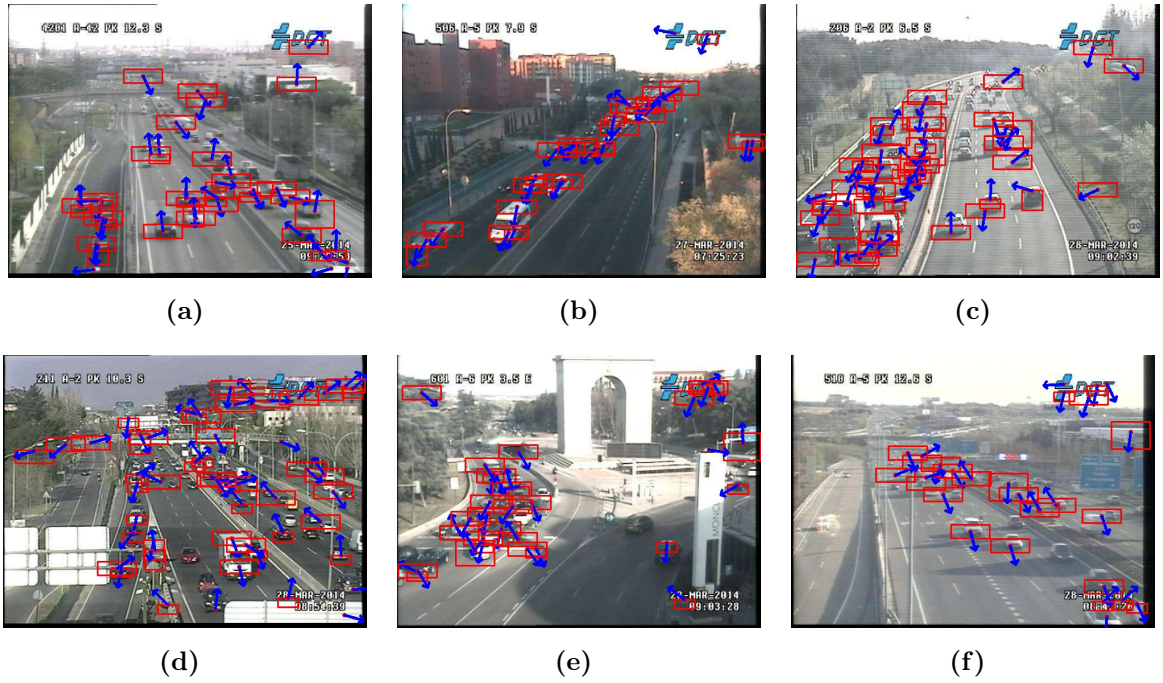
Tabla 3.1: Análisis del rendimiento de conteo de vehículos.

Ranking	Método	GAME(L=0)	GAME(L=1)	GAME(L=2)
1	Lempitsky+SIFT [17]	13,76	16,72	20,72
2	Fiaschi+RGB Norm+Filters [17]	17,68	19,97	23,54
3	HOG-2 [17]	13,29	18,05	23,65
4	Nuestro	11,85	18,32	25,25

lugar, lo que no es un problema ya que el valor del error es muy próximo (menos de 5-2 coches de diferencia en imágenes con alta congestión de vehículos) a los otros 3 planteamientos, centrados exclusivamente en estimar el número de vehículos y no su pose. Además, cabe destacar que este pequeño mayor error no es relevante para la tarea que concierne a este estimador en cuanto a la detección de carriles, ya que se demuestra que este no está lejos de las mejores estimaciones de centros de vehículos lo cual, le permite sin problema estimar los diferentes flujos de tráfico que permiten modelar los diferentes carriles de circulación presentes en las imágenes de tráfico entrantes al sistema.

Por otro lado, atendiendo a los resultados cualitativos, se ha podido deducir como la resolución de las imágenes de la base de datos TRANCOS [17] es suficiente para obtener unos buenos resultados en lo relativo a la estimación de centros de vehículos, pero no lo suficientemente adecuada para estimar la pose que caracteriza a cada uno de ellos, lo que es la base para una adecuada estimación de carriles.

La principal causa para que falle el estimador de pose, es que la base de datos TRANCOS, tal y como puede observarse en la Figura 3.8, contiene imágenes de congestiones de tráfico, donde los vehículos muestran un elevado grado de solape. Esto produce que aunque se estimen en la mayoría de los casos los centros de los vehículos de forma correcta, el estimador tiende a mostrar errores en la estimación de la orientación de algunos vehículos detectados, ya que éstos aparecen truncados u ocluídos, aspectos que dificultan tremendamente la estimación de pose.



(a) (b) (c)

(d) (e) (f)

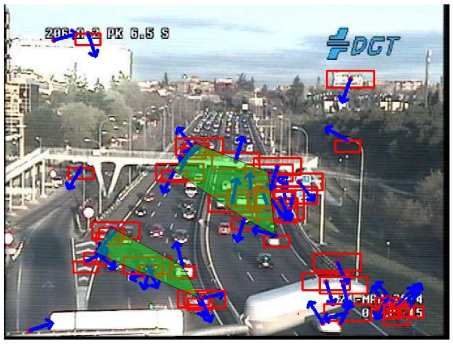
Figura 3.8: Estimaciones realizadas sobre varias imágenes de la base de datos TRANCOS.

En cuánto a la estimación de carriles, se puede observar en la Figura 3.9 como en la mayoría de casos, se producen errores en la estimación de carriles, dado que se realizan estimaciones superpuestas dentro de una estimación ya presente, como es el caso de las Figuras 3.9(c), 3.9(d) y 3.9(e). Esto, así como la unión de la estimación de varios carriles (como ocurre en las Figuras 3.9(b) y 3.9(e)), es debido a los errores cometidos en la estimación de la orientación en las imágenes de esta base de datos, dado que las imágenes disponen de una baja resolución (640x480 píxeles), y los árboles han sido entrenados con imágenes de vehículos de mayor resolución. Dado que la detección de carriles depende tanto de los centros detectados que se encuentren próximos, así como de la orientación de los vehículos próximos, si se cometen errores en la orientación en vehículos situados cerca del carril de circulación contrario, pueden aparecer los errores comentados.

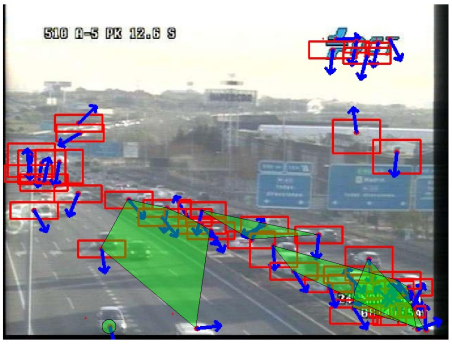
3.2.3. Resultados en la base de datos generada

Como decíamos, se ha generado una base de datos propia, compuesta de 213 imágenes con una resolución de 1689x942 píxeles, que mejora notablemente la estimación de pose de los vehículos, lo cual facilita las posteriores tareas de detección y representación de carriles de circulación. A continuación, se muestran una serie de fotogramas con los centros y la orientación de pose estimada de estos vehículos.

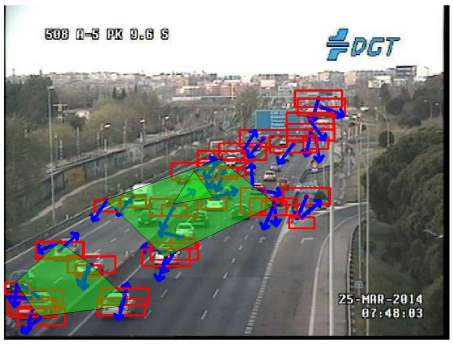
Como se puede observar en 3.10(c) y 3.10(d), en aquellos vehículos situados en la parte inferior de la imagen, hay mas posibilidades de que se produzcan múltiples detecciones en un mismo vehículo. Esto tiene su explicación en el factor de escala empleado para la



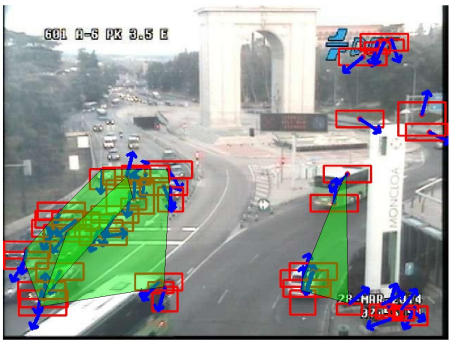
(a)



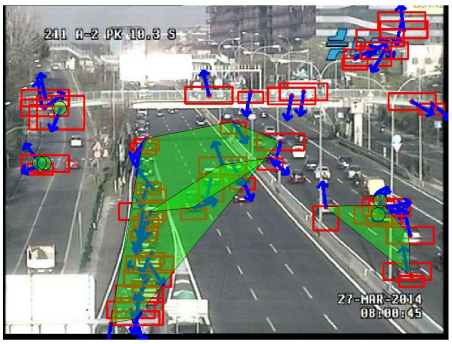
(b)



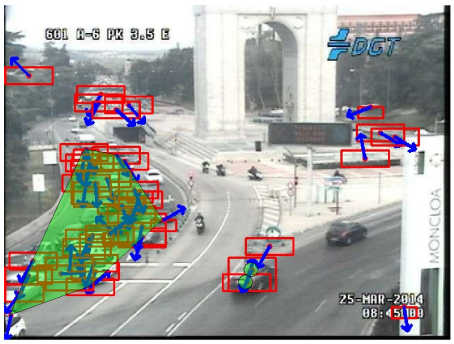
(c)



(d)

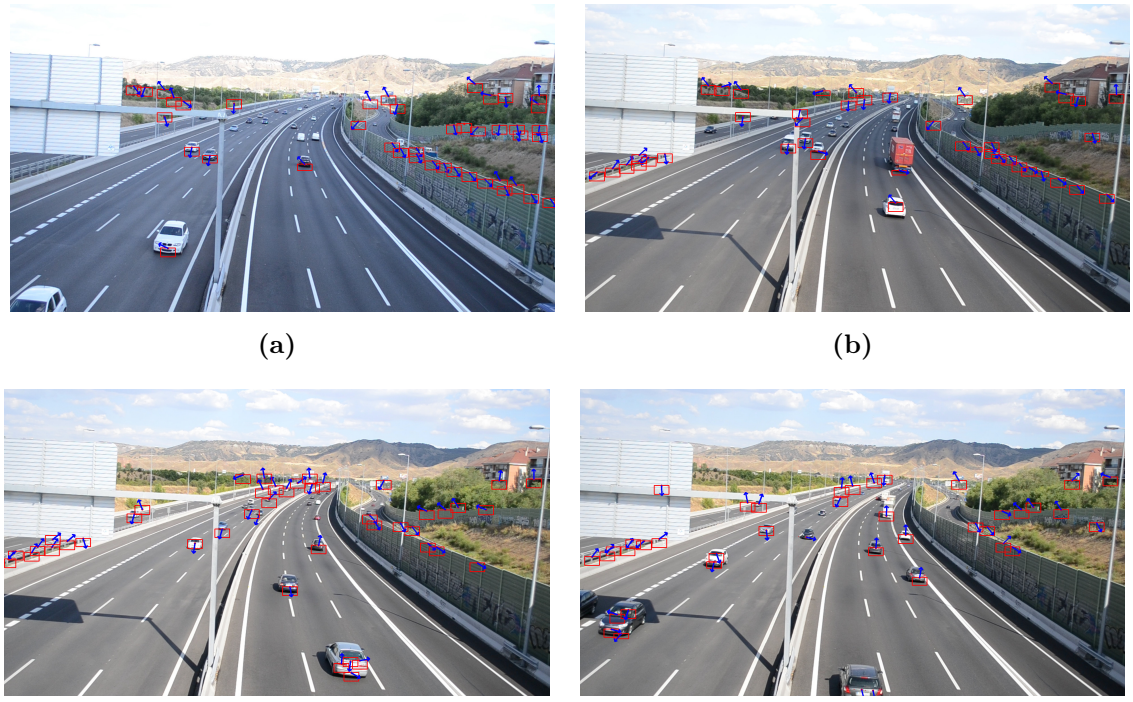


(e)



(f)

Figura 3.9: Resultados obtenidos para la detección de carriles en imágenes de la base de datos TRANCOS.



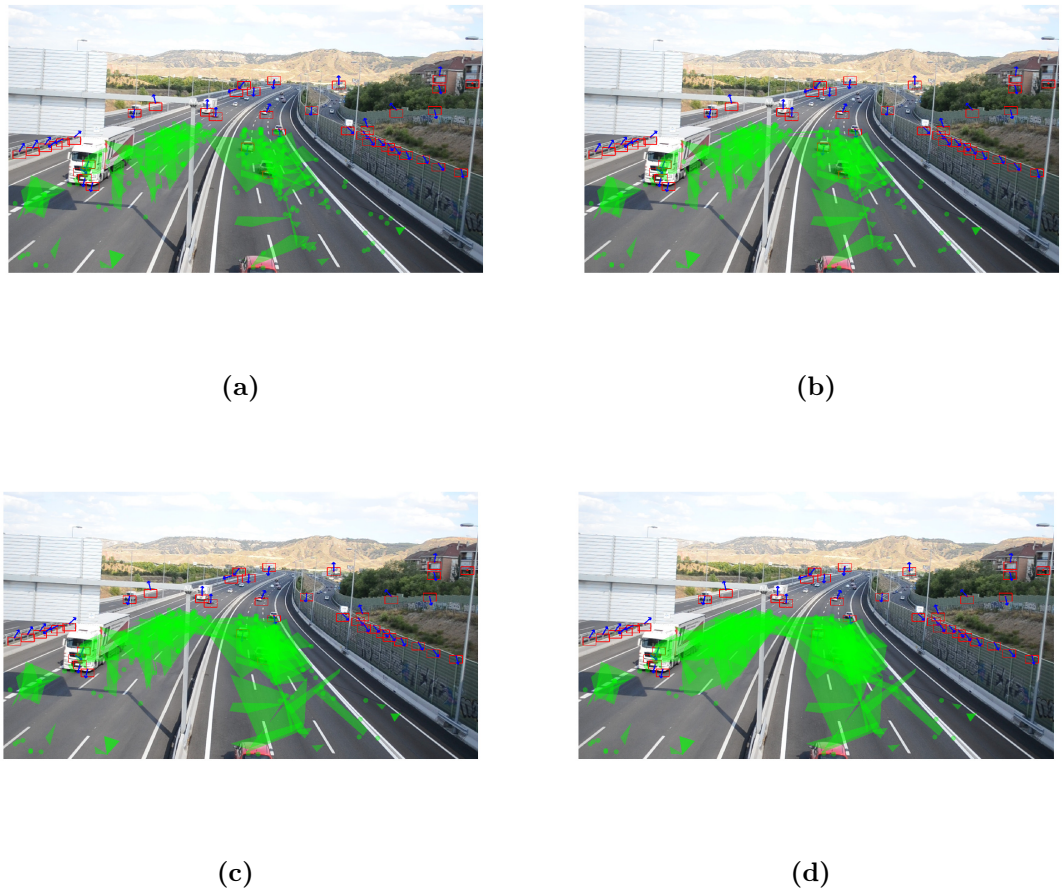
(a) **(b)**

(c) **(d)**

Figura 3.10: Detecciones de centro y orientación de cada vehículo presente en las imágenes de tráfico de la base de datos generada.

detección de los vehículos mas lejanos en la imagen, el cual es de un 1.4, lo cual quiere decir que cada parche lanzado a través de los árboles de estimación es agrandado un 140 % con respecto al que sería su tamaño inicial. Se ha elegido este factor de escala dado que ha proporcionado los mejores resultados en cuanto a detección de centros y estimación de la orientación de los vehículos a lo largo de los procesos de ajuste realizados durante el testeo del estimador en este campo de aplicación.

Por último, para realizar la representación de los carriles se ha hecho uso de la técnica de clustering jerárquico, explicada en la Sección 3.1, de un modo acumulativo y teniendo en cuenta tanto la distancia euclídea entre los centros de los vehículos estimados, como el ángulo del zenit que será tenido en cuenta por el algoritmo para valorar si diferentes vehículos se encuentran circulando en el mismo sentido o no. El término acumulativo implica el almacenamiento de las estimaciones obtenidas en una imagen, procedente de la misma cámara de tráfico, para disponer finalmente de la estimación completa del recorrido del carril. El interés de emplear esta técnica acumulativa, radica en el hecho de que no en todos los fotogramas obtenidos, se dispone de vehículos ocupando la calzada completamente, por lo que dependiendo de la imagen, la estimación de carriles podría resultar insuficiente. Esto se ha realizado ya que en determinados fotogramas la agrupación presente de vehículos en uno o varios carriles puede encontrarse ubicada en una zona concreta por lo que el carril representado queda solo presente en tal zona, viéndose el



(a) **(b)**

(c) **(d)**

Figura 3.11: Conjunto de imágenes separadas en instantes de tiempo no consecutivos donde se acumulan las detecciones estimadas para formar la estimación de carriles.

resto de la vía sin estimar.

En conclusión, se puede observar como la técnica de clustering jerárquico haciendo uso de la distancia euclídea, como medida de referencia para la agrupación de centros estimados y orientación de cada vehículo, es bastante eficiente a la hora de formar los diferentes carriles estimados, con el único problema de la perspectiva que forma la imagen en el fondo de la vía, donde la distancia entre vehículos se ve acortada visualmente, y por tanto se produce una agrupación ligeramente errónea entre vehículos próximos entre sí, que se encuentran en diferentes carriles en la zona de fondo de la imagen. A pesar de ello, visualmente se distingue de forma clara tanto la ubicación como orientación de cada carril, con lo que esta detección es sobradamente válida para cualquier soporte a tareas de conteo y videovigilancia de vías. Por tanto, se puede afirmar que los diferentes carriles pueden ser diferenciados de forma clara por el usuario, por lo que se dispone de una buena estimación de resultados, pero se podría mejorar empleando técnicas computacionales que modifiquen la perspectiva de la imagen, proyectándola sobre un plano, con el fin de evitar

el problema de solape de estimaciones en las zonas mas alejadas de la imagen.

Capítulo 4

Conclusiones y futuras líneas de trabajo

Este Trabajo de Fin de Máster ha consistido en el desarrollo de un estimador de pose capaz de detectar centro y orientación en cabezas humanas, incorporando un bloque de tracking en el sistema, para refinar la precisión del estimador, ya que se evita la presencia de detecciones ruidosas que puedan dificultar su implementación en una aplicación real. Además, se ha preparado el sistema para ser capaz de realizar la estimación de pose en vehículos, procedentes de imágenes de tráfico, para una posterior construcción de un modelo de detección de carriles. Esto se consigue agrupando aquellos vehículos cercanos entre sí, y caracterizados por una estimación de pose similar en términos de acimut, y el fin de obtener esta detección es la de facilitar tareas de videovigilancia y estimación de vehículos, ya que se presentan de forma clara los flujos de vehículos presentes en la imagen.

4.1. Estimación de pose de cabezas

En esta aplicación se han conseguido alcanzar unos niveles de precisión bastante notables. Esto se puede afirmar, a partir de los resultados cuantitativos presentados en la sección 2.4.2, donde se observa como nuestro modelo obtiene los mejores resultados en términos de estimación de centro y orientación entre los modelos de detección de imágenes RGB. Además, tanto en detección del centro de la cabeza como en la detección de alabeo, se han conseguido los mejores resultados incluso comparando con los modelos de detección en imágenes de profundidad, cuya facilidad para obtener una mayor precisión es superior, dado que estos estimadores deben analizar menos canales de características a la hora de realizar el proceso de estimación, con lo que sus estimaciones serán más sencillas, así como más ágiles. Además, atendiendo a dicha tabla, se puede observar como el nivel de fotogramas perdidos, que hacen referencia a aquellos fotogramas cuyo centro detectado se encuentra excesivamente lejos del centro real de la pose, se acerca en gran medida al 1 % presente en los modelos [27], [11] de estimación en imágenes de profundidad, ya que con nuestro 1,4 %, superamos al resto de modelos ya sean de estimación en fotogramas

de profundidad o de color. Por otro lado, también se puede afirmar que se ha construido un sistema ágil, además de preciso, dado que se ha conseguido una alta precisión en los resultados de las estimaciones, mientras se están realizando detecciones a una velocidad de hasta 3 FPS, lo cual puede facilitar la implementación de nuestro sistema en un campo de aplicación real, evitando así mantener el estimador en un ámbito puramente teórico. Como futuras líneas de trabajo, se propone la investigación en diferentes técnicas de segmentación de piel, con una menor sensibilidad a la iluminación presente en la imagen, con el objetivo de incrementar aún más la seguridad de que la imagen que será lanzada a través de los RF, será exclusivamente de la cabeza humana, sin presencia de ruido alguno, que en ocasiones puede producir falsas detecciones por parte de nuestro sistema. Logrando esto, la incorporación del sistema a sistemas de detección de fatiga o en mecanismos de interacción humano-máquina, sería totalmente factible y revolucionaria, proporcionando unos niveles de precisión y velocidad de funcionamiento muy altos, a la vez que se debe tener en cuenta un coste mucho menor al hacer uso de esta tecnología con respecto a las presentes en estos campos de aplicación actualmente.

4.2. Detección de carriles de circulación

En cuanto al rendimiento del sistema dentro del ámbito de la estimación de pose de vehículos y la posterior detección de carriles, se puede concluir el hecho de haber abierto una línea de trabajo que proporciona unos resultados adecuados, teniendo en cuenta el ser los primeros en realizar la detección de carriles mediante la agrupación de vehículos con posiciones cercanas y semejantes orientaciones. Se ha podido demostrar como el detector es capaz de mantener unos niveles de conteo de vehículos muy similar al que presentan otros sistemas, cuyo fin primario es precisamente el de conteo de vehículos, por lo que se puede afirmar que casi la totalidad de las detecciones realizadas sobre la imagen van a ser de utilidad para la posterior construcción de los diferentes carriles de circulación. En términos de rendimiento en cuanto a la detección de carriles, se ha podido observar como el hecho de haber entrenado los HF con imágenes de vehículos de alta resolución, dificulta una correcta estimación de la orientación en vehículos de imágenes procedentes de cámaras de la DGT, cuya resolución es notablemente menor (640x480 píxeles), por lo que el detector no es capaz de escalar lo suficientemente bien dichos vehículos como para estimar correctamente la pose de cada vehículo. Aunque también es cierto que las imágenes de TRANCOS no se prestan para evaluar un sistema de estimación de pose, debido a la gran oclusión que presentan los vehículos, aspecto que dificulta esta tarea de forma considerable. Por otro lado, tras construir nuestra propia base de datos de imágenes de tráfico, de mayor resolución (1689x942 píxeles) que las de la DGT, se han obtenido mejores resultados en cuanto a la estimación de la orientación de cada vehículo, y por tanto, también se ha podido obtener una detección de carriles de circulación mucho más clara y precisa, tal y como muestran los resultados cualitativos. Aunque éstos también

muestran la debilidad que presenta el sistema en la detección de carriles cuando, haciendo uso de la agrupación jerárquica para separar en diferentes carriles de circulación, ésta no proporciona el resultado esperado en las zonas más alejadas de la imagen, debido a la perspectiva de la misma, dado que en las zonas más lejanas los vehículos tienden a estar más cerca de los presentes en el carril contrario, por lo que se puede dar el caso de observar agrupaciones entre carriles diferentes en zonas alejadas de la imagen. En cuanto a las futuras líneas de trabajo, que se proponen para mejorar el rendimiento de la detección de carriles de circulación, se propone el entrenamiento de un bosque aleatorio con imágenes de tráfico cuya resolución sea similar a las de las imágenes de la DGT, para posibilitar la inclusión del sistema en tareas de videovigilancia en este ámbito. Además, con el objetivo de evitar los problemas de la detección de carriles en zonas alejadas de la imagen, se propone el uso de técnicas que proyecten una imagen sobre el plano sin perspectiva, para que la distancia euclídea entre vehículos sea equiparable en todas las zonas de la imagen. En concreto, la solución técnica pasaría por implementar un algoritmo que permita obtener una vista de pájaro de la imagen, de modo que el clustering se haga sobre una vista en la que se haya desaparecido la perspectiva. Una vez se perfeccione esta detección de carriles y se posibilite su uso de forma completamente funcional en imágenes de la DGT, se podría implementar el detector en sistemas de videovigilancia de la DGT para facilitar su labor, haciéndola más ágil y precisa dados los buenos resultados de los que parte el estimador de carriles.

Bibliografía

[1] J. Shotton A. Criminisi and E. Konukoglu. Decision forests: A unified framework for classification, regression, density estimation, manifold learning and semi-supervised learning. *Foundations and Trends in Computer Graphics and Vision*, Vol. 7:150, 2011. XIX, 8

[2] Kaehler A. Bradski, G. *Learning OpenCV: Computer Vision with the OpenCV Library*. O'Reilly Media, 1st edition edition, October 2008.

[3] L. Breiman. Random forests. Technical report, Statistics Department, University of California, January 2001. 7, 10

[4] M.D Breitenstein, D. Kuettel, Weise T. Gall J. Van Gool L., and H. Pfister. Real-time face pose estimation from single range images. *CVPR*, 2008. 4

[5] T. Cootes, G.J. Edwards, and C.J. Taylor. Active appearance models. *PAMI*, 2001. 4

[6] T.F. Cootes, G.V. Wheeler, K.N. Walker, and C. J. Taylor. View-based active appearance models. *AMFG*, 2000. 4

[7] M. Demirkus, D. Precup, J.J Clark, and T. Arbel. Probabilistic temporal head pose estimation using a hierarchical graphical model. *ECCV*, 2014. 4

[8] H.X. Ding and C. Fang. Head pose estimation based on random forests for multiclass classification. *ICPR*, 2010. 4

[9] Dantone M. Fossati A. Gall J. Van Gool L. Fanelli, G. Random forests for real time 3d face analysis. Technical report, Computer Vision Laboratory, ETH Zurich, 2012. XVII, XIX, 4, 5, 10, 11, 19, 21

[10] Gall J. Van Gool L. Fanelli, G. Real time head pose estimation with random regression forests. *IEEE Conference on Computer Vision and Pattern Recognition*, 2011. 4, 5, 10

[11] Weise T. Gall J. Van Gool L. Fanelli, G. Real time head pose estimation from consumer depth cameras. *German Association for Pattern Recognition*, 2011. 4, 5, 10, 21, 41

[12] Lempitsky V. Gall, J. Class-specific hough forests for object detection. *IEEE Conference on Computer Vision and Pattern Recognition*, 2009.

[13] Yao A. Razavi N. Van Gool Luc. Lempitsky V. Gall, J. Hough forests for object detection, tracking, and action recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 33(No. 11):15, 2011. 7, 9, 10, 14, 20

[14] A. Ghodrati, M. Pedersoli, and T. Tuytelaars. Is 2D information enough for viewpoint estimation? In *BMVC*, 2014. 4

[15] R. Girshick, J. Shotton, P. Kohli, A. Criminisi, and A. Fitzgibbon. Efficient regression of general-activity human poses from depth images. In *ICCV*, 2011.

[16] Galun M. Alpert S. Basri R. Shakhnarovich G. Glasner, D. Viewpoint-aware object detection and continuous pose estimation. *Image and Vision Computing*, 2012. x, 31, 32

[17] Torre B. López R. Maldonado S. Oñoro D. Guerrero, R. Extremely overlapping vehicle counting. *IbPRIA 2015*, 2015. 30, 33, 34

[18] A.S. Huang. Lane estimation for autonomous vehicles using vision and lidar. *Massachusetts Institute of Technology*, 2010. 5

[19] M. Isard and A. Blake. Condensation - conditional density propagation for visual tracking. *IJCV*, 1998. 5, 14, 17, 18

[20] M. Jones and P. Viola. Fast multi-view face detection. *Technical Report*, 2003. 4

[21] C. Ledbetter and K. Hui. Kinect for xbox 360. revolutionizing how we interact with machines. Technical report, Microsoft, 2012.

[22] L.P. Morency, P. Sundberg, and T. Darrell. Pose estimation using 3d view-based eigenspaces. *AMFG*, 2003. 4

[23] E. Murphy-Chutorian and M.M Trivedi. Head pose estimation in computer vision: A survey. *PAMI*, 2009. 4

[24] T. Nelson and Dailey D.J. Algorithms for estimating mean vehicle speed using uncalibrated traffic management cameras. *University of Washington*, 2003. 5

[25] K. Ramnath, S. Koterba, J. Xiao, C. Hu, I. Matthews, S. Baker, J. Cohn, and T. Kanade. Multi-view amm fitting and construction. *IJCV*, 2008. 4

[26] C. Redondo-Cabrera, R. Lopez-Sastre, and T. Tuytelaars. All together now: Simultaneous object detection and continuous pose estimation using a hough forests with probabilistic locally enhanced voting. *BMVC*, 2014. ix, 4, 5, 7, 8, 9, 10, 11, 14, 19, 20, 21, 26

[27] G. Riegler, M. Ruther, and B. Bischof. Hough networks for head pose estimation and facial feautre localization. *BMVC*, 2014. 4, 21, 41

[28] S. Schulter, C. Leistner, P. Wohlhart, P.M. Roth, and H. Bischof. Alternating regression forests for object detection and pose estimation. *CVPR*, 2013. 4, 5, 21

[29] M. Storer, M. Urschler, and H. Bischof. 3d-mam: 3d morphable appearance model for efficient fine head pose estimation from still images. *Workshop on Sub-space Methods*, 2009. 4

[30] Bennewitz M. Behnke S. Vatahska, T. Feature-based head pose estimation from images. *International Conference on Humanoid Robots*, 2007.

[31] V. Vezhnevets, V. Sazonov, and A. Andreeva. A survey on pixel-based skin color detection techniques. *GraphiCon*, 2003. 11

[32] S. Li H. Pauly M. Weise T., Bouaziz. Realtime performance-based facial animation. *ACM International Conference on Computer Graphics and Interactive Techniques*, 2011.

[33] G. Welch and G. Bishop. An introduction to the kalman filter. 2006. 5, 14, 17

[34] Movellan J.R. Whitehill, J. A discriminative approach to frame-by-frame head pose tracking. *Automatic Face and Gesture Recognition*, 2008.

Universidad de Alcalá
Escuela Politécnica Superior

ESCUELA POLITECNICA
SUPERIOR

Universidad
de Alcalá